

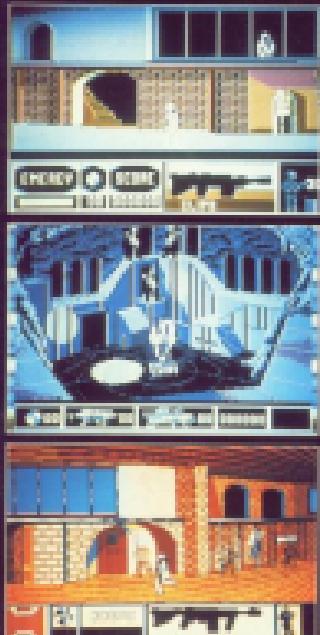
YOUR
COMMODORE

NOVEMBER 1989 £1.00

COMMODORE

AMIGA • C64 • PLUS 4 • PC • C128

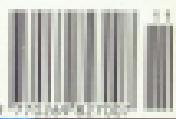
Comic Watch
Neon Zone
Rik of the Journos



MOONWALKER

'YOUR PROGRAMMER' Pull-Out

ISSN 0264-8277



DATTEL ELECTRONICS

THE ULTIMATE
GRAPHICS
PACKAGE...



SPECIAL
OFFER
PACK!



INCLUDES FREE MOUSE
MATHOLDER WORTH
£12.99



COMMODORE 1351 MOUSE

- ✓ 1351 Mouse is a high resolution two button mouse featuring optical encoding, button pulses, microswitches, rubber coated ball and high quality interface.
- ✓ When combined with OCP Advanced Art Studio this graphics package is quite simply the best system available. The features are unmatched...
- ✓ Create as images - objects, text, repeat patterns, etc.
- ✓ 1600 x 1000 pixels, 16 bit color.
- ✓ Repeat patterns or shades, mask, flood fill, zoom and magnification options.
- ✓ Save in as many as five modes.



VIC 20 RAMPACK

- ✓ Simple plug-in memory expansion cartridge for your VIC 20.
- ✓ Offers 16K of memory expand.
- ✓ Two models available - 16K or 32K.
- ✓ Both models are available for various memory settings 4K, 8K, 16K, 32K, etc.
- ✓ Many VIC programs require extra RAM and this is now easier.

ONLY £14.99 FOR 16K

ONLY £19.99 FOR 32K

HOW YOU CAN
INCREASE
THE MEMORY IN
YOUR VIC 20...



- ✓ Plug in RAM cartridge gives you up to 128K extra memory!
- ✓ Ideal for over 1000 new games and over 1000 new BASIC programs.
- ✓ Plug in RAM cartridge gives you up to 128K extra memory for your Commodore 64.

ONLY £14.99

COMPLETE WITH

COMPLETE WITH

OCP ADVANCED ART STUDIO™

**ONLY
£29.99**

TOTAL PACKAGE PRICING
1600 X 1000 16 BIT COLOR
INCLUDES FREE MOUSE MAT

EPROM ERASER

- ✓ This handy 100V 800mA device will erase up to 4 chips quickly and easily.
- ✓ Built-in timer and safety switch on power.
- ✓ Very safe - small enough to fit in pocket - easy to use.
- ✓ Works from built-in battery.
- ✓ Includes 100V AC adapter.

ONLY £39.99

LIMITED QUANTITY WHILE
AVAILABLE - LAST FEW

256K SUPERROM EXPANDER



- ✓ Allows instantly from 16 sectors to 128 sectors up to 1024 sectors.
- ✓ Programs, you can SUPERROM using your OCP programs.
- ✓ The need to load lots of cartridges just makes a selection from the SuperROM.
- ✓ Fully auto switch on power up.
- ✓ Instant swap after cartridge removal.
- ✓ Instant memory protection feature will make your programs harder to copy or steal from other machines.
- ✓ SuperROM requires no memory expansion.
- ✓ Requires 100V 800mA adapter.
- ✓ One board supporting over 1000 programs.

ONLY £39.99

EPROMMER 64™



- ✓ An easy quantity, easy use and effective programme for the VIC 100.
- ✓ Fully automatic, no need to worry about the order of the chips.
- ✓ Instant programme loading.
- ✓ Instant programme saving.
- ✓ Instant programme changing.
- ✓ Instant programme erasing.

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

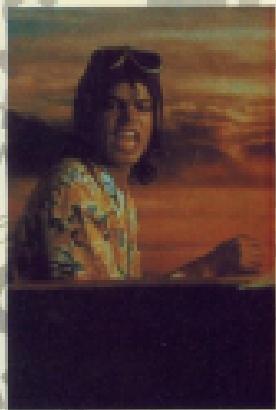
✓

✓



THIS
MONTH
PROGRAM

Left: Commodore
64 game
page 30



THIS
MONTH
PROGRAM

Left: Debbie
Clegg - page 31



THIS
MONTH
PROGRAM

Right: Steve
Moore - page 32

THIS
MONTH
PROGRAM

Left: Debbie
Clegg - page 31

YOUR COMMODORE

THIS
MONTH'S
PROGRAMME

THIS MONTH'S PROGRAMME

54 Bits of the Journals
The first of our INC. Photostories. No water-
resistant bag until he
entered the testing
highlights zone.

55 Printer Review
The Smith 24 - easier to
programme your computer
than a garage.

56 Arcades
They say 65m-ups can
make you go blind, John
Cookie would disagree!
Read his see this
statement.

REGULARS

52 Modernville
Habitat. He's a saint. He's
downright heroic worthy
Michael Jackson-like.
Habitat regular again.

53 News
And the mainpoints
again...

54 INC. Corner
Our monthly police-chat-
line.

55 Your Programme
The pull-out section with
full listings from the

GAMES

16 Curse of the Aztec
Bonds
Take heed all batters out
there. Do not touch the
Aztec Bonds!

13 Crystal Wing
Soon to be followed by
Capercornleg.

15 Citadel
Mausa Hattery

Titanic (think about that
one).

56 Big Wobbly Computer
Win a shirtless 'n' tie
signed by Bigwobbly
Henderson, Cope, Paul
Georgelina, horsey!

57 Software for Sale
Hope, it's not the
naughty lingerie pages,
but a serviceorient you
can purchase past and
present listings on disk
and tape.

14 Sword of Aragon
The pencil programme
than the sword. Unlike
of course, you're in the
middle of a medieval
batture.

16 Black Dangerous
And Adrian Dangerous
We are the dangerous
brothers...

17 New Zealand Story
Let's just hope that it's
not going to be a giant
Australasian sooty

58 Background
Plus 8, owners, we
haven't forgotten you.

75 Mailbag
A full-blown letter.

76 128 Corner
What do you call an
object with 128 corners.

52 Back Page
Are there no bounds to
what a scandal can
be? Scandalous, evidently
not.

18 The Soccer Squad
But what an offence
playing football? Oh
sorry, I thought you said
"Squad".

20 Xyborg
That's what you get if
you sit on a scorpion
boards.

21 APB
What are you going to do
with that truncheon,
Officer?

Autopilot Action
see page 23.

YOUR
PROGRAMMER

The contents of this publication including all articles, designs, drawings and programs and all copyright and other intellectual property rights thereto belong to Argus Specialist Publications. All rights are reserved by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications and any reproduction requires the prior written consent of the Company. © 1989 Argus Specialist Publications, 8 Leigham Court Road, London SW16 2HZ. Printed by Chase Webb, Plymouth. Opinions expressed in reviews are the opinions of the reviewers and not necessarily those of the magazine. While every effort is made to thoroughly check programs published in reviews we cannot be held responsible for any errors that do occur.

Year Commences incorporating
Year 6 is a monthly magazine
appearing on the first Friday of
each month. Argus Specialist
Publications Editorial
and Advertisement offices, Year
Commissions, Argus House,
Boundary Way, Shoreham
Boulevard HP2 2ST,
Telephone: 0423 90024.
Subscription rates upon
application to Year Commissions
Subscriptions Department,
10001 Lind, 3 River Park Drive,
Berkhamsted, Herts HP4 1HL,
U.S.A. Subscription Agent: Blue
Orb Worldwide Publications,
4335 West 10th Street,
Torrance CA 90506 U.S.A.

ISBN 0299-9277



SPEAKETH...

On Sunday 27th August 1989 I was invited to participate in the computer industry event of the year. Gransham Entertainments had hired Luton Town's football ground for the day, and ran a charity soccer tournament in aid of the Hillsborough disaster fund and the baby care unit at Liverpool Maternity Hospital.

Needless to say that although the journalists came running up on the day, due to a rather exciting penalty shoot-out, both Paul Kavanagh and myself thoroughly enjoyed our day out and we look forward to competing next year. Apparently over £3,000 was raised on the day and congratulations must go to Stephen Hall of Gransham and everybody else involved in the organisation for making it a successful venture.

This brings me nicely to the subject of football related games. Although we shall be exploring these in much greater detail in a later issue, I just find it quite startling that this year so many licensed soccer games are being released.

There's the Liverpool, Arsenal and Saint and Gransha games from Gransham; Gazzza's Super Soccer from Impulse; Manchester United from Krasil; and Super League Boys from Audiologic (being programmed with advice from Lou Macari). As soon as Vicent Jones' boxing comes out my life will be complete.

Look after yourselves until next issue and don't let your skin-pads slip...
Rik Henderson

Game Show Goodies

The jingle of Christmas bells will come early this year as the tiles at the Personal Computer Show ring in the new year. Panai will no doubt be quickening the hearts of the programmers and managers throughout the country as the 25th of September deadline approaches.

Although there are always a few surprises which are kept under wraps until the last minute, a few materials are thrown to the press to whet the appetite. So far Panai is fitting its materials to carry off the award as the most prolific software house of the decade. The company has lined up no less than 40 new releases for launch over the next seven months.

Fully animated aliens have been intercepted by the Panai team and bound into a 16 bit text called *Project Aerospace*. Also from the Interceptor Group, the low price label, Premier Judge, debuts its latest Joe Blaik epic.

There's always an international flavour to the Show and high quality software is the latest Spanish resort. Microdigital Soft is launching Dynamic's PC version of *Mary Meyer* plus three new titles: *Grand Prix Master*, a

Star Performers

Star has extended its range of LC screens with the addition of three A3 landscape format machines, one sporting a bottom feeder (Can we say that to our readers)?

The LC-15 is a wide carriage (16.5 inches) version of the fast, new LC-10 II. Both machines are developments of the phenomenally successful LC-10 but now with go-faster stripes added to their performance (80ops in NLQ). The LC-15 is the one which is also available as a bottom feed

new *Freddy Hartley* album called *South of Marfa*, and *After the War* a post nuclear holocaust entertainment.

Andrew Hewson will be hoping for a better Show this year after being dismissed by his star programmers on the eve of last year's exhibition. Amongst the games on preview this year is *Surgeon*, John Philips' follow up to *Elektrocar* and *Notch*.

CDS has stepped up upgrading to the latest computer denominator with a range of games from US-based Artwork. Ironically dubbed as 'adult software', the first two games are *Scrap Poker* and *Convoyfield Squares*. Wrist acting action for those who got their jollies from eight-bit daddies.

Mindscape's new games comprise *Snatch*, *Freddy Davidson*, *Freddy* and *Life or Death*. The latter is a blood and guts operation which places the surgeon's electronic knife into the player's hands - should hit the right vein for high scores but it may cost you an arm and a leg.

Touchline: The Personal Computer Show, Earl's Court, London, 21 September-1 October (Dodd only: 28-29 September).

version. Similarly the LC24-12 is a wider, faster version of the LC24-10.

Although Star Micros' classic LC series printers are low-cost business machines, the range is evidently suitable for home use. The prices start at £1399 for the LC-10, £1999 for the LC-10 II and LC24-10, £259 for the LC-10 colour version and rise to £3999 and £4999 for the LC-15 and LC24-15 respectively.

Touchline: Star Micros, 100 Uxbridge Road, Ealing, London W5 2JG. Tel: 081 560 1869.

Art of Tea-leaving

It's difficult to understand what someone is saying when their tongue is placed firmly in their cheek, however, we understand that Electronic Art's *Art of the Thief* promises to steal the Show this September.

Designed for Amiga and PC users who find *The Bard's Tale* too serious and monastic, *Keef* has his feet planted firmly on the ground as he creeps around Tri-city looking for a few takings. Thieving comes easily to *Keef* as he searches for the six pieces of the magic sword which will make him the new God King of Tri-city.

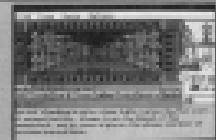
First, *Keef* must find the



Keef the Thief, meets Fred the Head, Sun the Fox, and Paul the Fox

scrolls which form a spell book. So, where's the magic about you know that *Keef* gets the scrolls - or maybe he always likes them.

Bard's Tale fans can take Art with the latest EA fantasy roleplaying adventure, *Swords of Twilight* (Amiga only). Written by Frostfall Associates, authors of the excellent *Archon*, the idea is to beat the living twights out of the guardians of the



Keef the Thief, meets Fred the Head, Sun the Fox, and Paul the Fox

seven enchanted swords and thereby depose the evil Shadcholoids who have subjugated the neighbouring kingdoms. As is life, the characters have memories so be nice to the people on the way up - you're bound to meet them again on the way down.

Touchline: Electronic Art, Langley Business Centre, 11-19 Station Road, Langley, Slough, Berkshire SL3 7TJ. Tel: 0875 514942.

Winter Draws On

Blue Ribbon hope to take the fizz out of a poor predictor program at the abominably low price of £2.99.

Systems 8 - The Pool Predictor is a dedicated database which the user gradually builds up as each result is announced. The program then takes the data and predicts the likely outcome of future matches in all four English and three Scottish leagues.

Hang on a minute, if hundreds of people buy the software and all enter the same data week after week, surely the prediction will be



the same for everyone and won't that reduce the share of any winners? Personally, we'll stick to our lucky pins.

Systems 8 is available for C64, C16, Plus 4, Amstrad CPC, Atari BBC Electron and MSX (who?).

Video Rentals

You've read the magazine reviews, seen the advertising, now you can buy the video. In a rather interesting move Action Screenplay is supplying rental versions of its publicity videos of the latest home releases to computer stores and video stores.

Now hardened game addicts will be able to thrill to the latest game demos with the added benefit of a video magazine. Rates in

the comfort of their living (or) room. The magazine features chart talk and areas of interest to amateur game players for a mere £1.50 rental charge.



Touchline: Matt Disk Entertainments, 107 Berkhol Road, Colchester, Essex CO4 3AB. Tel: (0206) 751415.



Yo Gazzza what's the score

Paul Gascoigne, the cross-prince of soccer, has signed up to a new team. But if you are the Spares fan don't worry, he is not leaving Totskicks, instead Gazz has signed up for Empire Software and will star in Gazz's Super Soccer.

The game is due for release in October on Amiga and C64 and, if you believe the hype, carries more features than there are calories in a Mars bar. Top of the list is the "foot-o-meter". This gives the player total control of the kick he is about to play, height strength as spin.

The pitch view will be different. With a number of variants depending on the position of the ball, though Empire say "it will work



well".

What if the Spares fan does not own a computer? Fortunately the game can be on for most of the 92 league sides and Scottish Premier Teams. Although I am sure it would be easier to play

with Gazz, than against (unless your name is Gary Lineker).

If the game has any of the personality of the Mars Bar Kid it should do well and may even top the league, unlike Gazz's other team.

Star Wars Trio

Domark is to release in three games based on the Star Wars films. The Star Wars Trilogy pack in the cover-up classics *Star Wars*, *The Empire Strikes Back* and *Return of the Jedi* for the Amiga (£24.99), Commodore 64 cassette (£12.99) and C64 disk (£19.99).

New from Domark the

home computer version of Tengen's own-up new game *Hard Drive*; from US company Broderbund comes *Shaggy's Cafe* and an action extravaganza against some hot alien competition and *Fascination*, an affiliation graphic designer which breaks Domark into the utility program field.

Hard Drive will be

available on November 23 for the Amiga at £19.99, PC at £24.99 and for the C64 for £9.99 (cass) and £12.99 (disk). *Fascination* is already in the shops but only for the Amiga and PC (£19.99).

Touchstone: Domark, Ferry House, 51-57 Lucy Road, London SW7 1JP. Tel 01-730 2221.

Dungeon's Drag On

A year after the launch of the C64 and PC version of the Advanced Dungeon and Dragons spin-off *Point of Resistance*, US Gold's SSD division has promised that the game will appear for the Amiga before Christmas - do they mean Christmas 1989, they surely do? So what's the release date? Dunno!

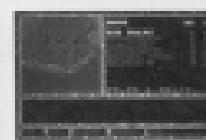
However, when the release date actually arrives, US Gold promises that the third scheduled AD&D game, *Blithfyre*, will be released one week later.

And? But what about the second game? Oh this the

company is almost more specific. *Dragon of Flame* will be appearing in late October for the PC and Amiga (£24.99) and in late November for the C64

(£14.99 disk, £9.99 cassette).

We look forward to this with anticipation but we're not holding our breath until then *Return of the Jedi* will suffice.



Go-ing for Gold

It's all systems Go for a top to bottom for Switzerland's Andrea Kierulff, the winner of the first Ami Computer Olympiad which was held on 13 August at London's Park Lane Hotel. All Kierulff had to do to win was to program the most challenging version of the popular oriental game of Go.

Each year Acer sponsors the world's computer Go championships in Taipei, Taiwan and the Olympiad was devised partly as a qualifier for the event. Conceived and organised by David Levy, an International Chess Master, the Olympiad also invites entries for other events such as Chess, Scrabble and Backgammon and this year attracted 85 programs from 17 countries.

In the Olympiad, programs compete against each other in a death or play battle to the death and the medals are awarded to the winners in each category. In the 1989 Go event the competition was expected to be stiff because the prize means a trip to compete in the world final, the winner of which can then try his program against the skills of a human expert. If the game wins, the programmer stands to claim a £1 million prize so Kierulff stands to gain a lot with his *Stone Explorer* program.

Any computers or board-game whiz-kids who think they can beat the world's experts can get further details from David Levy at the address below. The 1990 Ami Computer Olympiad will be held at the same London venue from September 8, August and 14 August 1990.

Touchstone: David Levy, 17 London Road, London SW3 5YF. Tel 01-91464356.

SANTA CLAWS

IS COMING
TO TOWN

**THE COMMODORE
CHRISTMAS SHOW**

YOUR MISSION IS TO GET THERE.
SANTA CLAWS IS OUT TO STOP YOU.



"BE THERE AT THE NONTEL HOTEL,
HAMMERSMITH, BETWEEN
17-19TH NOVEMBER 1989.
WHATEVER HAPPENS, DON'T
MISS THE COMMODORE
CHRISTMAS SHOW!"

Cheat Santa Claws - buy direct
from our Dealers!
Come and see ...

LATEST GAMES
EVER POPULAR C64
AMAZING AMIGA GRAPHICS
SOFTWARE AND PERIPHERALS
VIDEO AND AUDIO EQUIPMENT
FREE DEMONSTRATIONS

Look out for special deals
and Christmas bargains!



TOUR PRICE	ON THE DOOR	DISCOUNT
Adult	£25.00	£24.00
Under 16	£13.50	£13.00
Family (2 adults & 2 up to 16 years old)	£55.00	£53.00

Return this coupon to: Commodore PRESPORT,
Electronics Park, South West Lane, BOURNEMOUTH.

Please send me Family (2 adults & 2
children) tickets
for the Commodore Christmas Show 1989.

I enclose a cheque/PO for £ Cheque No.

Please stamp my Access/Visa card

Card No.

Expiry date Signature

One Yes No Other (please specify)

Postcode

Address

Postcode

Credit card users: 012 362 2662

C Commodore
WE'LL HELP YOU SEE THINGS DIFFERENTLY

CURSE OF THE AZURE BONDS

Curse of the Azure Bonds continues the series of games based on the Advanced Dungeons and Dragons role playing system and starts with parties of 5th level and over.

SSI - PC £29.99, C64 Disk £24.99

CURSE of the Azure Bonds continues the AD&D tale that ended with a great victory over the dark forces in the city of Phlan. Now with a party of 5th and 6th level characters you thought you were ready for anything. Unfortunately you were wrong, as you now find you party coming round in the city of Thieves and all is far from well. All magic and equipment is gone and now you have five blue symbols just under the skin of your sword arm. These you will soon learn are the handiwork of the less evil leaders of the New Alliance. Each one leads you to do their will again and your only way out of this curse is to destroy the New Alliance.

Your party of 12 brave adventurers can include up to six 5th level characters and can be loaded in from a Pool of Radiance of 100s for sure shot, although you may want to roll some new ones as existing characters could have race level restrictions. Or you may want to try out a Paladin or Ranger as these can now enter the fray. Luckily your hijackers have left the party with some money to purchase new equipment but upon attack-

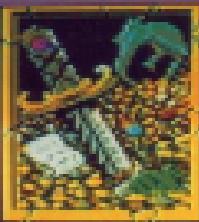
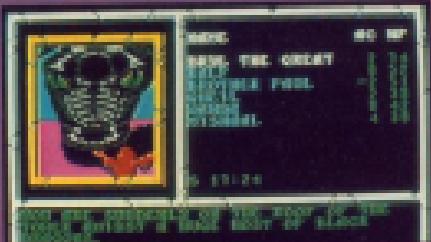
ing the high's carriage you find yourself thrown in jail only to escape in the sewers that have become the base for the Five Knights (a group of thieves). Here you will encounter your first challenge as one of your bonds matches their flaming blade symbol.

The game system will be familiar to those who have played Pool of Radiance (this game's prequel) and has the same 3D views, top down combat screens, slow but tactically accurate combat system and dazzling means to control your characters. However there have been some additions to speed up gameplay. The first is the FIX option included in the main menu that is used to take the pain out of maneuvering and casting "bare light weapon" spells. Select FIX and it's done automatically, although it does take up game time. In Pool you could spend ages wandering around the wilderness, but now a full ranged to try and find the best way to deal with creatures such as elementals, skeletons and beholders, as these are your new foes. You may find enemy clerics and magic users to be a problem but wait until you

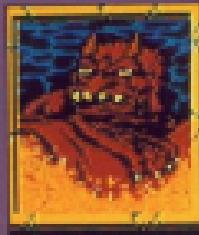
come across a mean little beast known as Dracolich which is an undead dragon that can attack with both a breath weapon - such as fire, cold or acid - and a paralyzing touch; it's just as well you can save the game everytime you enter camp. Consequently, the game has the census function with 30 areas that plagued Pool of Radiance and you face fewer more powerful creatures which makes for a more interesting game. The possible exceptions to this are the patrols of organised criminals, such as Dark Elves or Fire Keely thieves, but these follow set patterns and can be avoided.

As before, the on screen action is backed by references to entries in the accompanying adventurer's journal that also include tales and rumours that you can discover by clicking in the world map directs you to destinations via either trail, wilderness or even boat. Both of these speed up the game considerably and leave you to concentrate on the core.

Your problems are compounded by the various factions of evil that are fighting for the control of the Forgotten Realms, so it's safe to



Above: Don't you think graphics like these just drag on?



assume that almost everywhere is hostile and if you meet one of your "friends" things can get tricky.

Curse of the Azure Bonds also has a very different style of gameplay, as due to your attack, on the bing you don't have a fixed base to work on, so there's no systematic clearing of areas. Instead you must follow the trail of clues and rumours to confront each of your masters and their masters in turn.

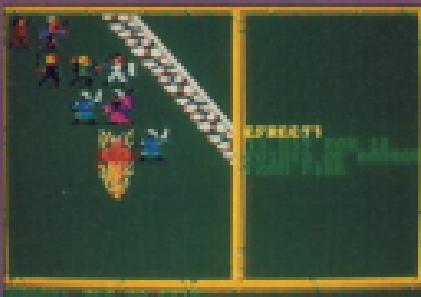
It may take time to adjust to combat with 5th, 6th, 7th and 8th level characters as you have so many other options to think about - such as which spells to use and how and when to attack. You'll also find it useful if you had an AD&D master local taverns.

There are few differences between the two versions of the game except that PC hard disk owners can do away with all the disk swapping associated with a game with 8 disk sides.

Curse of the Azure Bonds is the fourth game based on Advanced Dungeons and Dragons that began with Pool of Radiance. The good news is that there are more to come. Tony Hetherington

The AD&D games are the result of a three cornered deal between AD&D authors TSR, US software house SSI and US Gold. There are still four years of the first agreement left to run.

Below: Some of the amazing logo sprites.



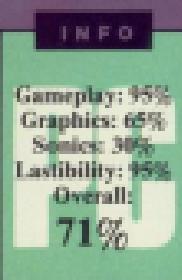
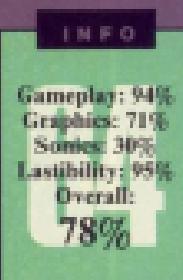
YOU SEE IT'S GOT UPGRADED TILBERTON GENERAL STORE

Above: (Sorry you right? You should never have visited Honest Bert's Tonic Emporium and asked for Bertie! and great).



SEEING AS HOW SSI DON'T, YOU NOTICE THAT SOME SPRITES ARE NOT SIMPLY UNTITLED WITH STUPID PATTERNS. THE REST OF YOUR PARTY ARE IDENTIFIABLE FRIENDS.

Below: If you ever want a high ranking army officer, Tilberton's the place to be!



GEMINI WING

Virgin - £9.99 Tape, £14.99 Disk

The ultimate in gutter press has started an intergalactic war with its jingoistic outpourings. Now you have to save the day.

Below: When you've killed the blob-like aliens...



Right... It's time to kill many more blob-like aliens!



This game features perhaps the worst scenario in the history of computer games. It is both unlikely and unrealistic and shows no regard for the laws of physics, logic or common sense. Unfortunately, it's the best part of an otherwise forgettable game.

The scenario creates around as Earth newspaper known as the SecondDay Spinn (I wonder what that could be?) and it silly stories about aliens turning girlfriends into potatoes and so on. According to Virgin these headlines had upset the aliens in question and when the SecondDay Spinn went too far with the heading "Die Mutant Alien Spins" the alien forces of every planet in the galaxy arrived to destroy the Earth. But wait! Is it a bird, is it a plane? No, it's you. Armed with only a single laser Gemini fighter you take off to destroy everything else in the galaxy to save the day and rack up a high score. Oh yeah? You have a much chance of winning big or leading England to victory in a Test series.

This takes you to the game itself which would be reasonable in a £1.99 budget

release but not as a full priced game. It's yet another about the aliens to collect extra weapons-style game. This apparently was caused by a "strange paradox in the fabric of reality" which probably also explains how Virgin can follow up a classic like Silkworm with a game like this.

Tony Hetherington

INFO

Gameplay: 12%

Graphics: 25%

Sounds: 38%

Usability: 10%

Overall:

21%

Back in the days of computer time Flight created a game about an Gemini Wing called Onshot. It was based on cricket. Unfortunately to change the name of a player, you had to break into the programme and change the code!

CITADEL

Activision - £9.99 Tape, £14.95 Disc

All was quiet, all was still. Darkness prevailed and neither good nor evil was present; until the monitor probes moved in to investigate an unknown power source.

For a time, nothing happened in the dark recesses of an underground complex on a distant planet. There was no light and no sound, and nothing moved.

Now had anything thought of moving for quite some time, but they were still charged up and, high above the planet's surface, a probe was watching, listening, monitoring and unidentified power source.

But the planet lay dormant. Nothing moved upon its surface, no birds flew and no fish swam in the seas, and yet, an electrical force was being detected.

Bringing the data back to base, the probe was replaced by a transport ship which landed five droids upon the surface. They set about finding the power and eventually traced it to a subterranean city.

Going underground, the probe began transmitting pictures of its surroundings and switch manual control for you to manoeuvre it around the mass of corridors which have revealed themselves.

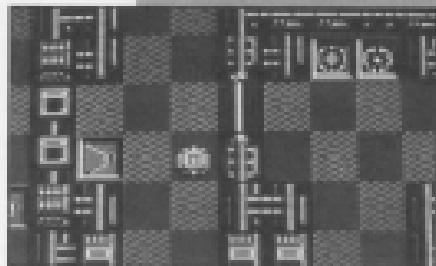
Cautious is the only approach in an unknown surrounding and if you do not observe this, you will become probes like all others on a hot day. As you near certain areas of floor, they open and gas tanks appear and open fire. Others open

and eject mobile death machines which roam around the maze and tack onto your trail. Following you they will launch projectiles at your probe and can seriously damage its circuits.

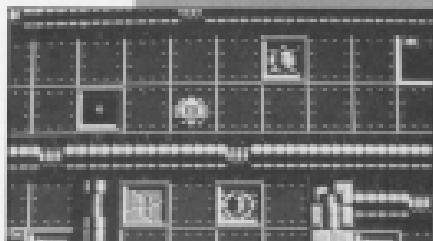
However, the design of these mobile robots has one major flaw in it: they are easily re-programmed using a standard code. Utilising this flaw, your probe can issue instructions to the robots which causes a change of heart and they rebel against their creator and protect you. Once re-programming is complete, they will still follow but they will get in your way and act as a shield. However, squabbles still break out if you decide to have more than one born-again robot following in your footsteps. Minions will run into the back of those already established and cause them to shudder.

There are various levels to navigate and each one contains a different breed of robots. Some seem to be armour plated and require more hits to destroy, others seem to be able to strip off the new programming and turn against you once more. And the dangerous bit is that you can't tell when they have lost their friendship.

Your energy level depletes as you come into contact with the enemy of



Above: I wandered slowly as a small mobile death...



Above: And now on... Mobility Squared is Mr. Bodd... Mortality Death...

their fire, but there are tanks which contain energy and fuel littered around the maze. Also on your travels you will find weapon pads which modify your distance. Lift shafts take you from level to level and transports zap you from city to city.

Ubiquitous sounding, great sound effects and decent graphics make *Citadel* one of the most playable games in quite some time. It's damned addictive and the best part is that it's not just a shoot-em-up and requires a few doses of sense and some strategic thinking to get you through. To Activision all I have to say is "please produce some more decent games like this".

Andrew Banner

Citadel is from *Electric Dreams* who produced such marvels as *Super Sprint* and *The Incredible Shrinking Sphere*. Marketed by Activision who have a recent track record of poor quality games. *Citadel* comes as a pleasant change.

INFO

Gameplay: 82%
Graphics: 74%
Stories: 83%
Lastability: 84%
Overall: 84%

84%

SWORD OF ARAGON

SSI (US Gold) - £29.99

Roleplaying, politics and war are all part of this fantasy epic.

The Duke of Aladda has died leaving you, his son, to your destiny. A destiny that demands for you to lead your people against the orc and goblin hordes and into battle with your enemies to the East. Only when you are liberator and leader of all Aragon will you fulfill your father's will.

Your first decision may be your most difficult as you must choose a character class to be throughout the game. You can be a great Warrior, Knight, Ranger, Priest or Mage - the last three making up their lack of fighting strength with their lack of fighting strength with magic.

Your choice of class also effects the type of armies you will raise as, for example, a Warrior can build infantry for half the cost of others.

Money plays an important part of the game as you must manage your city's affairs by setting tax rates and choosing between raising an army or developing trade and industry. Both have their priorities since you need an army to defend your city and expand, but you need resources to

develop, train and pay for it. You also have to remember that the townsfolk may not like paying taxes at 80% and may strike, leave or rebel, stopping your chances of success.

As Duke you must prove yourself as a leader and a statesman by dealing with envoys from other towns and resolving local disputes that can have an effect on your people's health, morale and loyalty. These in turn effect your revenue and the upkeep or expansion of your army.

Each turn represents a month of time in which you must manage your resources, defend your territory and expand to liberate the land. Which will inevitably lead to battles. When a battle ensues you must organise your forces for the fray. Obviously, your tactics will vary depending on the opposition you face and the units you have at your disposal. Personally, I favour battles that are won when an opponent's belief is reached you. Where they can be crushed by cavalry charges or by infantry armed with

swords, javelins or spears.

Priests, Mages and Rangers should be kept safe behind the line where they can use their magic to greatest effect. (Although you represent one class you can hire members of the others to supplement your army). Unlike other SSI games where magic users cast lightning bolts and hurl fistballs the magic in this game is more subtle. These spells effect the movement and fighting ability of friend and foe by changing the terrain or slowing an advance. Stamina can be restored or drained and actions can be healed, confused, levitated or disrupted.

At the battle concludes in turns (up to a maximum of 23) victory points are awarded for losses inflicted and territory gained, which finally decide the outcome of the battle. A loss will cause a drop in morale and loyalty but a great win will bring plaudited gold, new recruits for your army and experience for your troops. After a few battles these points will accrue to form increases in levels which will bring more

spells in your magic menu and better fighting skills to your units. For example, level 4 mounted bowmen cause more damage with every attack than level 1, 2 or 3.

The result is a fascinating game in which you gradually explore the resources and factors that you have and learn how to use them to expand your empire. The incredibly high Lascibility factor is well named as it's the style of game that you actually enjoy the more you play. Starting with a lost position when you're not quite sure what you're doing you gain military and character experience which adds you back to more and more. I'm afraid I'm not going to get much sleep until Aragon is free and I've earned the 600 points needed to complete the game.

Tony Hetherington



SSI is the company that took the headlines with the *Dragon's Crown* and *Dragon's Breath*. Games like this show why it got the contract.

149 F 9

Gameplay: 89%
Graphics: 56%
Sonics: N/A
Lastability: 95%
Overall:
91%



卷之三

THE BOSTON HERALD BOSTON 02108

On 27th Aug. of 1900, Davis, 1000 ft. above the sea, found breeding in the same tree.

第六章·第十一节

For more information, call 1-800-255-2264 or visit www.ams.org.

Printed from www.oxfordjournals.org

Answer: There's no positive result on an application for a new lot of land.

Volume 10 Number 10 October 2000

Digitized by Sarmi

www.ijmsi.com

THE JOURNAL OF CLIMATE

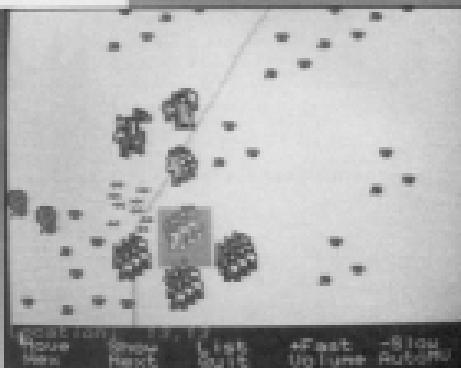
19. *Leucosia* (Leucosia) *leucostoma* (Fabricius) (Fig. 19)

THEORY OF THE STATE

1996-1997-1998

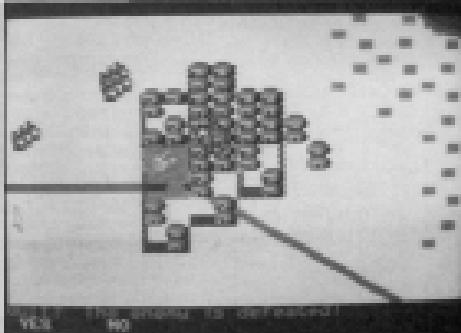
Successful completion of a high-level international conference

Please see [How to Download](#) for more information.



When you start a new job, you're likely to have to learn the ropes.

Volume 28 Number 10 • October 2000 • ISSN 0898-2603 • 100

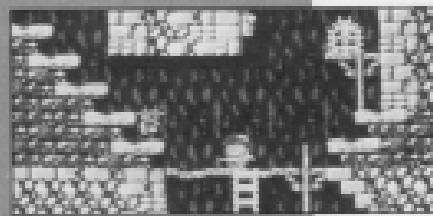




RICK DANGEROUS

Explore the temples and avoid the traps, but will there be a chocolate orange for you at the end?

Firebird - £9.99 Tape, £14.99 Disk



Above: But Rick, that's dangerous!

INFO

Gameplay: 65%

Graphics: 60%

Sounds: 50%

Lastability: 45%

Overall:

55%

They say that there is no such thing as a true joke: everything is a derivation of something else. To some extent, the same applies to computer games. Firebird's latest game, Rick Dangerous, shows distinct signs of a common ancestry with assorted platform games of many years ago.

Our spacyman hero, part-time explorer, part-time stamp collector and whole

time good guy has been on the trail of a long lost South American tribe for many a long year. By a strange coincidence, when his plane runs out of fuel over the jungle, he just happens to land at the entrance to the very Aztec temple he was searching for.

Getting in proved to be easy. Getting out at the other end was decidedly more problematical. No sooner had the deer slammed behind him than a huge stone ball starts to chase him down the passage. With no time to change your name from Rick Dangerous to Rick Moss (because as we all know, a rolling stone gathers no Moss!) Ha ha, very cool! Get on with it - Dangerous Rick! your only option is to throw yourself off the cliff at the end of the tunnel. Surprise, surprise, gravity works in South America too and the ball follows you...

Having eventually dodged out of the way of the evergreen concrete marble, all you have to do now is similarly avoid a series of angry natives, a few snakes and bats, lots of poisoners

spiked pits and a whole load of spars shooting out the walls, before you finally reach the other end. Then you can try your hand at three other scenarios including a mighty fortress and an Egyptian tomb.

In order to defend yourself, you can pole things with your stick. Timing is all important here and I found the technique useful only against bats. Native are best dispatched with a swift bullet but you only carry a few of these and the rocks they make is all too likely to set off a trap. You will also need dynamite to blast your way past rock falls and the like.

The trouble with this game is the lack of variety. Problems are usually only solved by trial and error - you frequently have to throw yourself into the unknown. Once you have solved that part of the game though, it is unlikely to cause you further stress and soon becomes repetitive. Fun and simple to play, it would suggest that it is overpriced somewhat. It would make a very good budget game.

Gordon Handlett

NEW ZEALAND STORY



Your chance to become a fully paid up member of the Antipodean rescue service.

Ocean - £9.99 Tape

On the face of it, birds are something of a failure. New Zealand's most famous bird, they are totally incapable of flight, not very good at running and pretty ugly to boot. They do have one redeeming quality though, they are extremely loyal. So when 20 of Tiki Kru's friends are captured by a psychotic wolverine, he doesn't hesitate. Armed only with his bow and arrow, he sets off to rescue his friends before they experience a wolverine's digestive system from the inside.

The game is a variation on the old ladder and platform theme, but is none the worse for that, having been skilfully converted from the arcade original.

You must reach all of your friends within a certain time allowance. A radar

shows your relative locations and, on the early screens, there are a few helpful arrows to point you in the right direction. Although you only have your bow and arrow to defend yourself with, shooting some of the enemy will force them to drop other, more potent weapons - bombs, homing missiles and pulses of temporary invulnerability being examples. Collecting fruit scores bonus points and if you manage to pick up all the letters of the word 'friends', you are granted an extra life.

Monsters - bunnies, bocceballing throwers, evil frogs and vampire bats will all impede your progress and there are also some very large guardian creatures that have to be defeated, not always by orthodox means - the only way to escape



from the whale is to allow yourself to be swallowed first. You are then caught up literally in a shoot out!

As well as shooting on land, you will also need to take to the air. As already mentioned though, your wings are not equipped for this purpose so you will need to hijack a balloon to help you. You will also need to watch your oxygen levels when swimming underwater. Drowning is not good for your health!

New Zealand Story is very well presented and is one of those rare games that although simple to play and easy to get into, is highly addictive. Above all, it is good fun which, when it comes down to it, is what a good game is all about.

Gordon Hamlett

Below: Hamlett Plays Kiwi, my favourite.



Below: If you need a new book, buy one from us. Two months later we'll send you the bill.

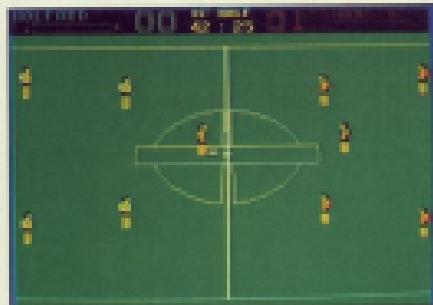
INFO

Gameplay: 95%
Graphics: 75%
Sound: 65%
Lastability: 85%
Overall:

80%

The Soccer Squad

Now that the new football season is well under way, the software charts are dominated by football games. Gremlin has fielded a team of four experienced players as its challenge for the top.



PLAYER STATUS	NUMBER OF
SHP	~1000
LST	~1000
SHD	~1000
SHD	~1000
4TH	~1000
-	~4000



Gremlin Graphics - £3.99 Tapes

Four games for the price of one is always worth a look as it often offers great value for money. This one bundles together Canadian football games starring two of the top names in football, Gary Linsker and Roy of the Rovers. However, before you think that all football games are the same, check out this as each one offers an unique challenge.

Gary Lineker's Superstar Soccer is the traditional style of football game, with a scrolling side-on view of the pitch and joystick controlled players. In the game you can take control of the centre forward, goalkeeper and manager - in which you can pick your team from a squad of players and decide tactics by telling forwards to shoot or pass, and defence to either stay back or support the forwards. Perhaps, the best part of the game is that the forwards don't just shoot at goal willy nilly they can also attempt spectacular goal attempts with headers and overhead kicks if the ball's at the right height.

Gary Lister's Super Skills is a selection of training exercises, that include gym training (such as sit-ups and squat thrusts), field work (including deliberately missing, chipping and shooting) and ball control - where you must keep the ball off the ground by bouncing it off various parts of your person.

Boy of the Year is a curious mix of football action and youth adventure, as some unscrupulous gang has kidnapped the Melchester Rovers team just before a big exhibition game that could save the club from

the developer's build-area. In a race against time you must avoid enemy supporters, hoodlums, traps and ambushes and find the other four players, otherwise you will have to face the opposition alone.

The fourth game is the intriguing Footballer of the Year which plays like a board game and casts you as a 17 year old just entering football. Through taking the limited scoring chances that fall your way in action situations you will earn money and be spotted by scouts from other teams, gradually taking you from fourth division obscurity to the division and international football, and perhaps the ultimate accolade as the Footballer of the Year.

Although together they represent a good footballing package, they are, at best, 2nd division players as they lack the quality and depth of the top of the game.

Tom Hetherington

Gremlin Graphics has now moved out of the US Gold stable and back to Sheffield and independence.

INFO

Gameplay: 90%
Graphics: 88%
Stories: 35%
Lastability: 42%

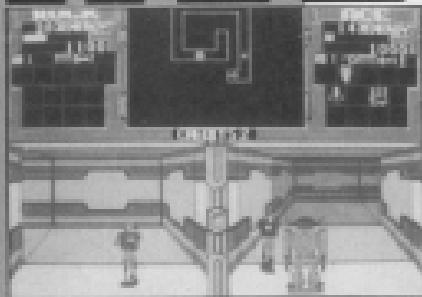
Overall

11 - 12

41

100

Xybots



Above: Aaargh! There's a robot coming at my XY bottom

Elect the heroes, two batch commando type characters who waddle birth to victory as though suffering from the after effects of a grueling A visitation.

Despite the ridiculous wobbles our heroes are given, if a little limited. As they move through the maze complex of the Xybots their progress is shown on a small, circularised section of the screen. This limit typifies two player games of this nature. Although it enhances a game so as to let two players to play simultaneously, the loss of playing

area often spoils the game play. And when nearly half the screen is taken up with a map, things become a little ridiculous. The only advantage in this game when playing with two players is the ability to shoot each other in the back and the combined firepower.

The Xybots are a mechanical race and not confined to the usual limitation of the human organic body. This means they can move a lot faster, carry more armour and pack twice as much power into their shots. Despite this advantage and your rapidly dwindling 'power

Demark - £9.99/£12.99

A mechanical menace in the form of the Xybots is assaulting the Earth. All that stands between them and victory are two waddling heroes.

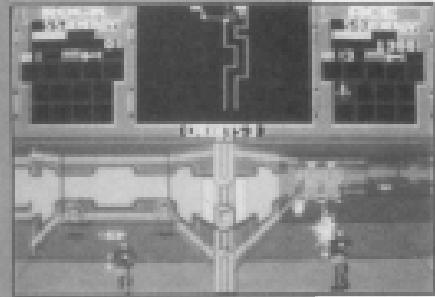
supply you still manage to slay the wretches. This could possibly be due to the artifacts picked up here and there on your travels. Coins can be collected for later purchase of weapons and extra firepower. Energy-capsules to keep the body functioning and a host of padpons to boost your characters abilities.

On level one a little ingenuity is rewarded with the discovery of a teleport pad that takes you directly to level eight. Not only does this allow you to skip the intervening levels but a cache of coins ensures that your character can purchase enough hardware to handle the challenge. If you decide to take the longer route you will be rewarded by a display

of hostile mechanics never before seen on your 64. Small robots fit through the maze moving collectible items, large screen loving robots guard the passage ways and the intervening droids with the flags, a killing blow can only be landed on these when their doors are open. The lack of colour makes the retiming also a little awkward to spot so just count the number of droids currently in circulation and if the number is greater than your double dining capacity remember to dodge.

Remember not to panic when a little sign pops up and points to same danger off to one of your sides, so to turn you need to press the fire button and move the stick; the other way around and you blast the wall.

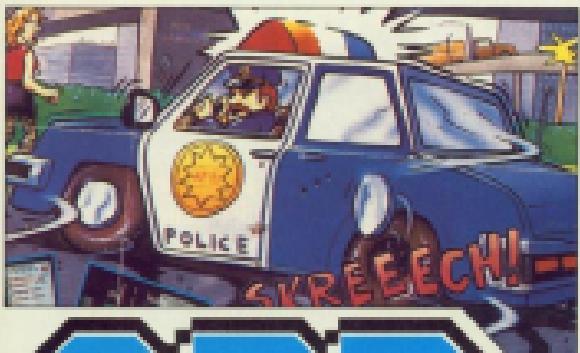
This is not so much a bad concession but a bad game which, without the advantage imposed upon it by a dedicated arcade machine, fails to produce goods. Adrian Pamphrey



INFO

Gameplay: 47%
 Graphics: 60%
 Sound: 78%
 Lastability: 73%
 Overall:
60%

Xybots is one in a range of Tongan conversions by Demark, who currently holds the license to convert all Tongan releases to home video format for the next three years.



APB

Domark - £9.99 Take £12.99 Back

The third game from Atari subsidiary Tengen puts you behind the wheel of a police car.

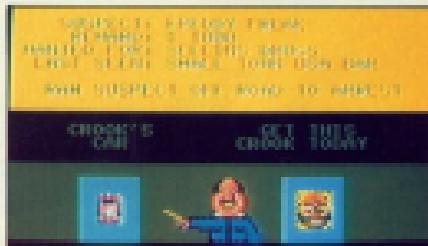
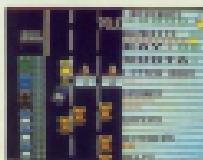
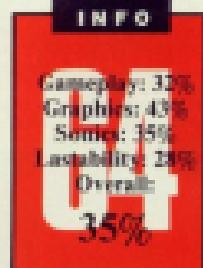
A PR, or All Points Bulletin, calls you Bob, the heroic officer Bob who must patrol busy streets, searching for criminals to apprehend. However, these crooks are far from dangerous, consisting mostly of litter bugs, drunkards and bimbos. You must arrest enough of them to meet your daily quota.

Your job is made easier by the fact that these crooks drive around in colour-coded cars so it's simple enough to drive up behind them, sound your siren to pull them over and bag yourself a bonus. To help you, you can also pull into a gas station for more petrol, grab a doughnut to give you more time and visit a speed shop to speed up your pace car by adding nitro, engine speed, acceleration, and even a gun to shoot more serious crooks.

By day three of your eight day course you will be skilled to track down a more serious criminal going under a subtle pseudonym, such as Fredddy Freak and Sid Sniper. He isn't as easy to track down as the litterhaga and has to be removed off the road before coming quietly. Once caught he must be interrogated by wagging the poxbox left and right in order to get him to sign a confession before your chief arrives.

APB promises cartoon style humor and graphics, but delivers a Spy Hunter style driving screen and tiny bumbling and repeat actions that barely fill a fraction of the screen. This is the third Tonga game where collecting credits, in this case money bags, can be cashed in at a shop for improvements to your tank, robot or

卷之三



Happy What Day with every thought

Wiley | Blackwell | Oxford University Press | John Wiley and Sons



SMOOTH CRIMINAL

If we were to believe everything that we read in our national dailies, not only would we have found the size of smooth, but we would be led to think that Michael Jackson has committed every strange act known to man, short of living on the moon.

Luckily most people have the sense to shrug such comments and try to understand the man as a human being. He was born in 1958 in Gary, Indiana, where he, and four of his brothers, formed a band which was to take the world by storm.

The Jackson 5 were first brought to attention in 1968 when they auditioned for Motown Records. And it was their appearance on the Ed Sullivan Show in 1969 that brought them firmly into the public eye. Michael was the lead

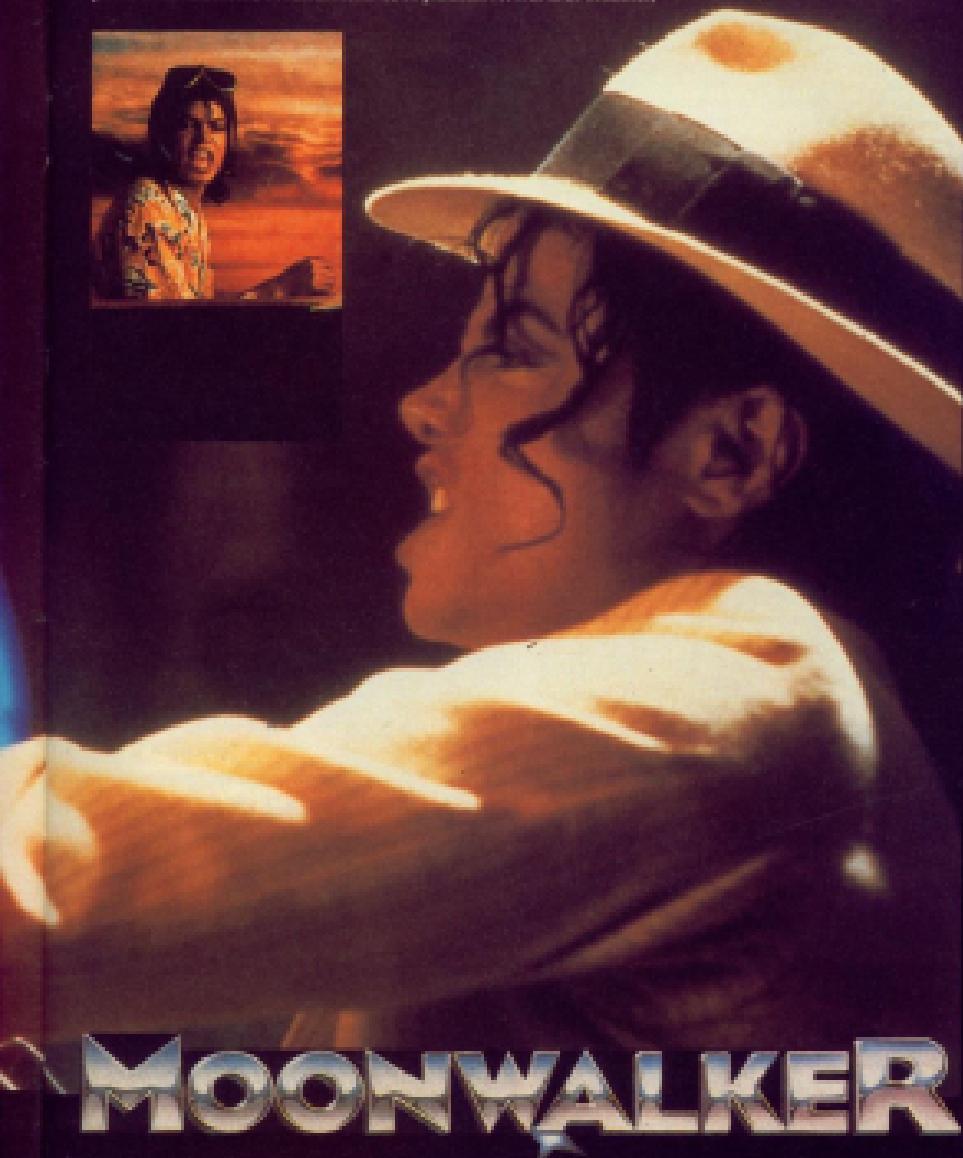
Michael Jackson, pop phenomenon or over-publicised eccentric? Rik Henderson explores the man, his music, his movie and previews the forthcoming game from US Gold.



singer, and was but 9 years of age when they experienced their first number one hit.

When Michael was 12 he recorded 'Beat', which was the first solo hit of many, and it was in 1978 that he split from his Family group to explore the world of music, and superstardom, on his own. This was after appearing in the pop remake of *The Wizard of Oz*, entitled *The Wiz*.

Although he was already at this point a star in his genre, the album *Thriller* placed him in the record books. He received 50 platinum records in 28 countries.



MOONWALKER

■ FEATURE

and in America he was awarded no less than 8 grammys. To date Thriller has sold over 40 million copies worldwide, and is still selling.

How does one follow that? Easy, one makes one's first feature length movie based on one's own records.

Moonswalker is Michael's first feature length film (having previously only done shorts, such as Thriller) and is titled after his best-selling autobiography — which is named after the dance that he has performed so well. It deals with all the subjects that Michael has shown as being dear to his heart — youth, friendship, drug abuse, bad journalism, and villainy — and in true fairy-tale fashion, it takes many obstacles and shows how to overcome them (esp. kids, this is all believable stuff). If you are confronted by a drug dealer and his troops, just transform into a large robot and blow the book out of them.

The plot follows Michael and three friends (one of which is Sean Lennox) through various subplots, all of which are orchestrated by Mr. Big — an evil drug baron with an evil haircut. First probe that Mike (as is the journalists at a Hollywood studio, they are hungry for blood and with no grubs some Jackson signature (any resemblance between these journeys and the euthanasia found at Argos is purely intentional).

He manages to elude them by wearing a rabbit mask, and in some scenes property, he speeds away on a motorcycle until he finally escapes their noxious pencils. This is not the only transformation that he undergoes in Moonswalker.

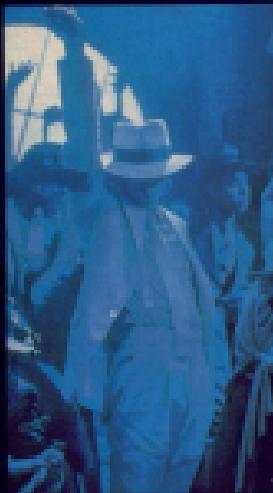
Like I said earlier, he also transforms himself into a very large Transformer-like robot, and even more unbelievably, a mammoth space ship, in which he finishes the battle with a few swift swerves of his thrusters. Moonswalker is definitely intended for kids, and has many themes which makes it ideal for such an audience.

It also makes it ideal for a computer game. Which is just as well as US Gold has secured the license and will produce Michael Jackson's exploits in time for November, and thus Christmas.

The game follows the plot of the film very closely and is split into 4 levels. The action is played in a maze-like game with a top-down view, although in-between each level there is a moderately large action sequence. It follows all the transformations, culminating in a VERY large action sequence featuring the Spaceship Jackson.

The music throughout the movie is also going to be converted onto the home computer, with relevant song appearing in relevant sections, and the whole game is going to be very monochrome. US Gold is very excited about this project and, with the help of the Keyspanch Corporation of Minneapolis, Moonswalker is likely to be a success. Worldwide (much like Michael himself).

If US Gold only sell 10% the amount of copies as the record Bad did, everybody up in Birmingham will be happy little bunnies.





**Andrew Brown, the man
around town, goes to
PCs.**

Welcome to PC Corner, the part of the magazine for all Commodore 64 users, whether fledglings or old hand. This month it's a mixed bag of mainly software reviews, plus an intro to batch files. First though, I'm going to kick off with a couple of news items.

Lotus corp (see Magellan review) has been busy lately. Not only has it got both 128 releases 2.2 and 3.0 onto the streets, but has upgraded the help available through its dealer network. The new system uses a CD-ROM disk called unimaginatively, CD/PROSPECT. With over 10,000 pages of information on the disc, that fine happened-race of '96 it's not in the manual, well have to check with head office as I longer builds any water.

With the intel 80486 chip now becoming available as upgrades for systems such as the IBM PS/2, and AST Premium 386 machines, speculation over processor upgrades for other machines is rife. It seems however that Commodore who only has 20MHz 386's is not yet to be tempted, or was it caught out? Certainly the much rumoured 386SX machine is on the stocks, but what else?

Batch File Fun

It is often thought that MSDos computers do not have a programming language built in. In fact they do, the so-called 'Batch language', which really is just an extension of MSDos.

Batch programs are just like those you may have written in Basic, but there are differences. The commands available are somewhat limited, and each program is really a text file. However, some useful things can be done. For example with Autostart bat. This is the Batch program that controls the computer at start up, so it's very important. Much can be done with

PC CORNER ■

**MSDOS
SOCD
A
S
,**

Autostart.hat, and I may cover the options in future columns, but for now here is a simple mod for you single drive users. The following text should be prepared with whatever editor you use, and saved as a text-only file:

```
echo off
path=.
prompt $P$G
copy command.com <
```

You should also prepare another text file with the following command:

```
File=20
Bytes=20
device=Ramdrive.sys
Set romope=C:\command.com
```

Next, make a working copy of your boot disk and copy both text files to it. Also copy the file Ramdrive.sys to the disk. Rename the first file Autostart.hat, and the second Config.sys, and everything should go as normal, except for a message similar to this:

```
Microsoft Ramdrive Version 2.01 Virtual disk <
Disk size 64k
Sector size 512 bytes
Allocation unit 1 sectors
Directory entries 64
```

What you have done is created a RAM disk which becomes drive C:, and copied Command.com to it. You may well be wondering what this will do for you. The answer is that you should no longer see messages like this: Invalid Command.com

Insert Command.com disk in default drive and strike any key when ready.

Command.com will always be available to the system, and irritating disk errors kept to a minimum. Hard disk owners can still use this trick incidentally, as it does speed things up somewhat. The one drawback of course being that memory is lost to the system. Note, this is only meant as an example and some programs will not work under this configuration of DOS.

Typographics

Just about everybody these days has heard of DTP, however fewer people actually put the theory into practice. Reasons for this are complex. An important factor has been the gap between low-priced low-performance systems, and the more exclusive Postscript based kits. Some might regard this state of affairs as a bonus, preventing the perpetration of typographic disasters on the world.

One way of narrowing the gap is to provide the low end user with some of the excellent fonts libraries only available to Postscript users. The G&T have done, with their Typographics range of outline fonts. The normal way of acquiring these is by purchasing one of the two collections. Prime, consists of Sans, Serif, and Courier which is the equivalent of Helvetica, Times and Courier on Postscript printers. Standard, has all the rest, and is much more interesting, with such things as Zapf Dingbats and Sans Narrow. Together they make up the equivalent of the 13 fonts available as standard on Postscript printers. The two collections come complete with a three ring binder, manual, disks and poster showing the fonts available. Individual fonts are also available.

If you want to make use of the fonts, you have to install them first. This is a clumsy process at best, and irritating at worst. Everything has to be specified from menus and sub-menus, you can't just wizze around the screen selecting items at will. Also, as the program generates bit-mapped fonts, the file sizes can be excessive. For example, I specified an extensive range of point sizes from 8 to 72 in a single style of one outline font. Typographics suggested 18 minutes 31 seconds to generate the files, which then occupied 234kbytes on my hard drive. Of this, the 72 point style occupied no less than 1.5Mbytes. Sample calculations then gave me a figure of 7.5Mbytes for the six fonts supplied as part of the standard collection. The moral has to be, use the minimum



number of fonts necessary to do the job, unless you have a massive hard disk. Unfortunately you are then severely hampered in your choice of fonts, effectively back to square one. And what happens if you wish to use the fonts with two or three different applications?

On the plus side, a large number of applications and output devices are supported including typesetting machines. These however can only be used if you specify the postscript names for the fonts when generating them.

My overall verdict; a cost-effective way of widening the scope of your DTP work, but really needs a large hard disk to get the most out of it.

Lotus Magellan

Magellan has been hailed as one of a new breed of DOS shells. What's a DOS shell you ask? Strictly speaking, they are a way of controlling DOS without using the system prompt. That includes everything from simple batch programs to MS Windows. A stricter definition however would add the fact that some part should remain in memory while your programs are running. This allows the main section to be reloaded when your program terminates. By that definition, Magellan is the Rolls Royce of DOS shells. It has elements of artificial intelligence with its 'fuzzy' search routine and 'Hyperedit' facilities that allow you to treat all the files on disk as one enormous database.

Magellan builds an Index of files stored on your hard disk, known from a list of templates what is in those files displaying the contents accordingly. Dbase and Lotus 123 files are just two types it knows about.

You can search for data in files in several ways using the Explore option. Magellan's file viewers allows it to display the contents of the files in a form near to that of the creating application. Thus Lotus 123 files are displayed in row and column format, Dbase files as tables, and so on. Launch (F7) is another powerful command, because it not only allows you to start up programs from within Magellan, but also gives extensive control over the process. It does this by making use of the powerful Macro facility built into Magellan. Of course it's not perfect, but it could save you a lot of key strokes over the year.

All the other facilities, such as copying, deleting, renaming and printing files are also present, and as you would expect from such a polished product are well implemented.

Documentation is probably the most surprising I have ever seen. There is a ring bound manual, plus two small booklets, and all rest of class. However I found the content pedestrian, if thorough. Full marks though for Index and Glossary, essential for this kind of product.

Magellan is a well finished program, but what will be its uses? Clearly Lotus intends the product to fit in with its current base of corporate 123 users, but who else? Certainly anyone who needs to organise their hard disk, or where a number of people need to access data on one machine. But, the kind of disorganized people who do need Magellan are, by definition, the first likely to purchase it.

EXCLUSIVE STEVE BELL INTERVIEW

ZEKE MANY WORLD MUS

ZEKE MANY WORLD MUS



Rik Henderson, the man in the dayglo tights, deals a severe dose of comic capers

comics, Lucy (adult, Cf or like comedy f. note, speech designed to amuse, distract, burlesque, satirize, C. song, paper, history of *Beano* incident), *2nd Floor, 1, Comedian*, "paper (house, paternal publication full of horrors, at-a-glance, math-practising, quasi, odd, offy u.l.t. Gk (Athenae reveal))

— The Pocket Oxford Dictionary, Fifth Edition

Although everybody must have a vague idea of what a comic is, I feel the dictionary entry above contains a phrase which sums up to definition perfectly. "Fictional publication full of horrors" may not say 'Beano' to you, but we're moving into the 1990s. And the comic industry is coming with us.

Long gone are the days when children flocked to the newsagents to see if Spiderman had vanquished Doctor Octopus (again) and if Superman had pulverised Lex Luthor (again). Now dark landscapes are the settings for brutal crimes and many superheroes keep their buster shorts inside their trousers. Batman has acquired weaponry (and lost a partner), Dan Dare (the great, great, great, great, etc. guardian of the original that he has died and been resurrected). And even Judge Dredd has turned from being a fearsome psychotic killer into an old fusty psychotic killer.

This can be put down to one thing: Comics are no longer for children only. A fact that became apparent when *2nd Floor* was launched in December 1979. The publishers now claim to sell around 800,000 copies every two months. The content is crude, Barbara, and at best can be described as toilet humour, which gives half way to explaining why 11 (and another 700,000 dedicated readers) love it so much.



Another recent comic which has conferred one famous character to a more adult status is 'The Dark Knight Returns'. Written by Frank Miller - who is, at this very moment, writing the script and supervising for *Blade Runner 2* - it looks at the future of Batman, and plays a fitting end to the caped crusader career. Many scenes from *The Dark Knight* were converted into comic form via Tim Burton's *Batman* masterpiece, thus proving what a strong plan Frank had devised.

But these were just the first of the 'mature' readers' comics. There was an enormous flood of them in 1987 and '88, and we can expect many more to come. One other of honourable note is 'Watchmen' by British comic author Alan Moore. It examines superstitions in a realistic society and comes to the conclusion that heroes is something lowered upon by a majority of 'real' people. Of course, like *Dark Knight Returns* *Watchmen*, Terry Gilliam has already signed on the dotted line to convert this piece of comic literature onto the cinema screen.

Both of these products came from America based DC Comics. There are of course many British comics that hold their own in the UK market. One such comic is 2000AD. Founded in 1977, it sought to replace *The Eagle* as a space-age action paper for young and old. Fortunately it did far more than that.

It has established itself as the top of it's kind (although competition is very scarce) and it's characters are now household names (in my household anyway!). As from its 650th issue, they should all become household names in America too, as it will be in the shops there at the same time as here.

Also under the 2000AD name is Crisis, a comic that seeks to educate whilst remaining enjoyable and not too heavy. Dealing with political and social values, it is incredibly well drawn and is a rare comic that will get me into a newspaper ready to receive a copy.



VIZ

PRINCE CHARLES NAMED IN BIZARRE

Another new magazine to come our way is *Deadline*, which is edited by two of 2000AD's greatest artists. It can be described as a more down-to-earth, populist version of 2000AD, more than anything else, and it seems to be doing extremely well. Having just held it's first birthday party at the Limelight (where London's West End), it contains several strips, such as the new influences 'Tank Girl' by Jamie Hewlett, and a few interviews and features on up-to-the-minute items. *Deadline* is monthly and is *massively* highly recommended.

While DC are looking to update their characters, Marvel haven't quite got the idea yet. That said, Marvel UK has recently released an amazing little comic called 'The Street Rascals'. This is a sort of *Blues Brothers* in the future. It follows the exploits of two private eyes in a city full of debauchery, crime, sex and violence (although not necessarily in that order), in a very humoursous and well worth seeking out.

The software houses have also decided to capitalise on the not-in-popularity of comics. Virgin Interactive have announced two games based on famous characters. *Silver Surfer* based on the Marvel superhero of the same name (he's silver and he's got a wacking great red board on which he can fly), and their second offering is *Viz*. They say that the latter will not be copied (so this may ruin the feel of the game, but the cover will display a sign to the effect of 'unauthorised game only').

Omega, of course, has *Batman* - the movie, but it also has the rights to *Wolverine*.

The Edge has *The Punisher* (a vigilante driven by his family's death at the hands of a crime organisation) and *The X-Men* (a superhero group containing many rough and ready characters). It has also announced a few more to come next year - *Barbarella* (a blind superhero who is anti-drugs and just about and everything else as well), and *Wolverine* (a character who has large, very sharp claws that he can extend from his knuckles).

Siemens has *Doctor Doom's Revenge*, which is based around the Fantastic Four's oldest and most feared enemy.

However, who says that there will be new games based on (main for all): Captain America, Judge Dredd, The Swamp Thing, Dr. Foster, The New Mutants, and Rogue Trooper. We can be guaranteed though that this will not be the end, that is a whole universe of license just waiting to be grabbed.

Roger Mellie

The Man On Telly





YOUR PROGRAMMER

CONTENTS

34 Sprite Positions

Automatic control for sprite positions, as they wander the screen.

37 VAT Number Checker

Is that invoice valid?

Check the VAT number printed with our easy-to-use program.

39 Moving the Screen
Relocate the screen in memory.

41 Special FX

Fade your text in and out, and produce effects for your programs.

43 String Along

Making use of the three \$ basic functions.

45 Expanding BASIC 8
The eighth part of our on-going series, as it nears its impending end.

48 Privokey

Keep your privates private with this security program.

50 The User Port

Use the user port.

How to use the pull-out...
Remove from magazine and cut spine. Punch holes where indicated and insert in a ring binder for easy storage and long-term protection.

If you want to achieve a truly three-dimensional quality to your sprites here is an incredibly useful routine for you to use. It's very short, so there isn't endless lists of data to type in.

Examples of three-dimensional sprites which spring to mind are the various space simulations, such as International Soccer.

As you probably know, sprites have a default priority arrangement in that the lower the sprite number

Make use of the priority properties of your sprites

By J. Simpson

SPRITE PRIORITIES

the higher is its priority. This means that sprite 0 has the highest, through to sprite 7 which has the lowest.

Sprites with higher priority always display in front of those with a lower priority. This is fixed within the hardware of the VIC III chip, which means if you want to create a three-dimensional illusion, then a routine needs to be constructed which will manage the sprites by keeping 'foreground' sprites higher in priority.

To handle this, I decided that sprites lower down the screen would be considered as 'foreground' to those higher up the screen. I'm sure you know that the pixel map is arranged with the 'Y' coordinate running from 0 at the top of the screen, to 255 at the bottom. The Machine Language routine, SPRITE PRIORITIES, uses the 'Y' coordinate information to decide which sprite should be where; the sprite with the greater 'Y' value becoming sprite 0, and so on.

Within the routines

1. SPRITE PRIORITIES this is the ML routine which deals with checking and updating all the sprite positions and their priorities. If a sprite moves up the screen (or backwards into the picture), and is doing so it passes above (or behind) another sprite, then sprite priorities will swap the two sprites around, together with all data relevant to each (i.e. Image, Mem, Xpos, Ypos, Colours etc). This neatly holds together the illusion that a sprite's priority is changing - due passing in front of, then behind another sprite. This does, however, pose a sticky problem. Let's say that the joystick is being used to control one particular

sprite, and the other seven are static. Their default values are:

- Block 1 = White
- Block 2 = Red
- Block 3 = Cyan
- Block 4 = Purple
- Block 5 = Green
- Block 6 = Blue
- Block 7 = Yellow
- Block 8 = Orange

You can use the numerical keys (1-8) to select any one of the eight blocks. That block will now be under CONTROL, and can be manipulated around the screen using the joystick plugged into port 2.

If you study the listing, you will see that in the INITIALISE routine (lines 68-79) variables 'Y' and 'K' have been declared, and the two arrays P0 and DS0 initialised. At line 76 the arrays are filled - P0 with 0 to 7 and DS0 with the 'Y' coordinate value of each sprite, from 0 to 7.

sprite element. Normally that control would be defined and controlled using one particular sprite - say, sprite 0. All that needs to be done is to pack and pass the ML, equivalently with 'Y' coordinate information into sprite 0. However, should sprite 0 move up the screen and pass the next sprite, then CONTROL shifts to sprite 1, and if CONTROL carries on up the screen it might become sprite 2, 3, 4, 5, 6 or 7.

2. BASIC DEMO - For programmers, and to show the system working, the basic demo program outlines a demonstration of SPRITE PRIORITIES in action, as well as useful routines for the controlling of which sprites are where. When you 'RUN' the program (that is after loading SPRITE PRIORITIES, and typing 'NINW', then loading BASIC DEMO), eight sprites are displayed diagonally across the screen. The

array P0 holds the current position of the BLOCK (not the sprite) on the screen, and DS0 - which derives from 'Dummy sprite' - holds the current 'Y' location of where each sprite would be, should there be no SPRITE PRIORITIES manipulation.

The variable 'K' is used to shift joystick CONTROL over the block selected from the numerical key input. A loop checks through the P0 array to find the current screen position in relationship to the sprite. For example, BLOCK 3 might be at the bottom of the screen, and so it would be sprite 0. Where 'N' is set to equal the actual sprite value - in the foregoing example, 0. When the joystick is moved up or down, program control will call either JOYSTICK UP (commencing at line 11) or JOYSTICK DOWN (line 12). Let's say 'up' is the selection. First DS0 is decremented (K-block being moved), and the screen

parameters are checked (line 22). Then the updated value in DS8(E) is poked into the 'Y' register of the sprite holding the data for that BLOCK. 'W' holds the sprite number.

Next, the CONTROL sprite's 'Y' coordinate is checked against the next lesser prioritised sprite 'Y' coordinate. If CONTROL is greater, then it maintains priority and so the program skips lines 23-27 and returns to MAINLOOP - no more needful to be done. However, should the coordinate value ever be less, then line 25 calls the ML routine SPRITE PRIORTIES, where priority and all relevant sprite data is swapped from one to the other. On return from the ML routine, 'W' is now incremented to the next highest sprite number. This is followed by an error trap, and UPDATE flag ARRAT (line 46), which will update PINS for the current sprite position. Missing down the screen in the reverse of the above - check out lines 12-19. Left and right have no effect upon priority, and so these are standard routines to Park and Poke 'X' coordinates.

A Stage Further

To take things a stage further and have multiple sprite movement on the screen, the variable 'W' would need to become an eight element array. Each image, block, or whatever, would be given a constant value from 0 to 7, and when the program updates 'Y' coordinate for each element, ML(Element Number) would be used. This could be followed by a line such as:

ON (ELEMENT NUMBER)
GOSUB (PARAMETERS)

Here subroutines would handle different images, or elements, and/or situations.

I have not incorporated a collision detect routine within the Basic demo as the main purpose is to show how effective SPRITE PRIORTIES is, and to offer a demonstration of a method of control. Collision detect should operate quite normal without any problems. For ML programmes, the conversion of the Basic demo routines into source should prove to be quite elementary.

Sprite Image Data

I have included a listing of data for sprite images which will display numbered (1 to 8), three-dimensional, coloured boxes. The numbers corresponding with numerical boxes and the colours as before. Should you decide to use these images, type in the basic loader and data lines, and save to tape/disk. You will now have to make some changes to the BASIC DEMO program.

First, delete line 78 entirely.
Change line 71 to read:
POKE P+C, PG+C
Remove the REM from line 72 and type:
POKEV = 28, 255
Remove the REM from line 73 and type:
POKEV = 17, 11
Remove the REM from line 74 and type:
POKEV = 38, 12

This enables the multicolour mode and sets the colours to Grey 1 and Grey 2.

Getting it all in

Sprite Priorities - Type in and save the basic loader program.

Sprite Image Data - If you are going to use the sprite data which is included, then type in and save this listing also.

Basic Demo - Type this basic program in. If you are going to use the additional sprite data, make the necessary changes outlined above and save it.

Running the Demo

First load and run SPRITE PRIORTIES then type 'NEW'. If you are using sprite image data then load this in and type 'NEW'.

Now you can load and RUN the BASIC DEMO program. Remember keys 1 to 4 will select the BLOCK you wish to control with a joystick plugged into port 2. Well, that's about it. I'm off to the pub for a pint. Hope you like SPRITE PRIORTIES, and can utilise it and expand upon it...

PROGRAM: SPRITE PRIORTIES

```

83 2. PRIMITIVES, ZONEAR, SPRITEPRIORTIES
  - LOAD INIT, CHECK INIT, DATA - --+
  84 3. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  85 4. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  86 5. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  87 6. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  88 7. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  89 8. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  90 9. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  91 10. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  92 11. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  93 12. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  94 13. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  95 14. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  96 15. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  97 16. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  98 17. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  99 18. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  100 19. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  101 20. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  102 21. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  103 22. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  104 23. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  105 24. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  106 25. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  107 26. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  108 27. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  109 28. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  110 29. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  111 30. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  112 31. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  113 32. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  114 33. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  115 34. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  116 35. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  117 36. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  118 37. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  119 38. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  120 39. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  121 40. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  122 41. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  123 42. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  124 43. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  125 44. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  126 45. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  127 46. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  128 47. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  129 48. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  130 49. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  131 50. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  132 51. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  133 52. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  134 53. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  135 54. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  136 55. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  137 56. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  138 57. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  139 58. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  140 59. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  141 60. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  142 61. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  143 62. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  144 63. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  145 64. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  146 65. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  147 66. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  148 67. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  149 68. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  150 69. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  151 70. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  152 71. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  153 72. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  154 73. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  155 74. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  156 75. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  157 76. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  158 77. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  159 78. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  160 79. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  161 80. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  162 81. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  163 82. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  164 83. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  165 84. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  166 85. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  167 86. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  168 87. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  169 88. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  170 89. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  171 90. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  172 91. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  173 92. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  174 93. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  175 94. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  176 95. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  177 96. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  178 97. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  179 98. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  180 99. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  181 100. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  182 101. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  183 102. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  184 103. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  185 104. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  186 105. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  187 106. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  188 107. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  189 108. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  190 109. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  191 110. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  192 111. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  193 112. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  194 113. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  195 114. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  196 115. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  197 116. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  198 117. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  199 118. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  200 119. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  201 120. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  202 121. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  203 122. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  204 123. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  205 124. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  206 125. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  207 126. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  208 127. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  209 128. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  210 129. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  211 130. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  212 131. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  213 132. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  214 133. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  215 134. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  216 135. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  217 136. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  218 137. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  219 138. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  220 139. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  221 140. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  222 141. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  223 142. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  224 143. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  225 144. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  226 145. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  227 146. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  228 147. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  229 148. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  230 149. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  231 150. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  232 151. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  233 152. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  234 153. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  235 154. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  236 155. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  237 156. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  238 157. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  239 158. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  240 159. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  241 160. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  242 161. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  243 162. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  244 163. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  245 164. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  246 165. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  247 166. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  248 167. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  249 168. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  250 169. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  251 170. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  252 171. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  253 172. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  254 173. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  255 174. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  256 175. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  257 176. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  258 177. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  259 178. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  260 179. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  261 180. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  262 181. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  263 182. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  264 183. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  265 184. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  266 185. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  267 186. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  268 187. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  269 188. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  270 189. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  271 190. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  272 191. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  273 192. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  274 193. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  275 194. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  276 195. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  277 196. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  278 197. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  279 198. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  280 199. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  281 200. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  282 201. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  283 202. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  284 203. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  285 204. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  286 205. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  287 206. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  288 207. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  289 208. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  290 209. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  291 210. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  292 211. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  293 212. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  294 213. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  295 214. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  296 215. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  297 216. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  298 217. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  299 218. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  300 219. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  301 220. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  302 221. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  303 222. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  304 223. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  305 224. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  306 225. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  307 226. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  308 227. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  309 228. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  310 229. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  311 230. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  312 231. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  313 232. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  314 233. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  315 234. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  316 235. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  317 236. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  318 237. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  319 238. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  320 239. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  321 240. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  322 241. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  323 242. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  324 243. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  325 244. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  326 245. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  327 246. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  328 247. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  329 248. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  330 249. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  331 250. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  332 251. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  333 252. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  334 253. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  335 254. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  336 255. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  337 256. SPRITE PRIORTIES, 128x128, 128x128, 128x128
  338 257. SPRITE
```

卷之三

PROGRAMMING -

VIA NUMBER CHECKER

Make sure your VAT invoice is a bonefide invoice with this simple checking program.

By R. E. M.

Have you ever wondered whether that 15 per cent VAT added to your bill

really does go go go the Tasmania? Are you worried that the plumber's bill, accrued in pencil on the back of a

cigarette packet, may not be genuine! Well, worry no more! With the aid of this simple test, basic necessary

you can now know for sure that the firms you deal with are decent, honest and true. Well, actually you can only use it to find out whether the VAT numbers they put on their invoices are genuine or not. But this does go quite a long way towards proving their credentials. You will, once you start using this program as a regular part of your book keeping routine, sleep more easily at night, safe in the knowledge that you haven't forced customers to pay over the odds for your goods, and that the Taxman won't be claiming you've cheated him with plenty of invoices.

The program is written in Basic, and will run on the C64, C128, Plus 4 and C16. To use it, simply enter the nine digit VAT number and press return. The program will respond with the appropriate message:

VAT number is valid

or:

VAT number is not valid

Simplicity itself! The program is foolproof, objecting to any input which is not a nine digit number. If such an input is attempted, an appropriate error message will be

printed before the program reports that the VAT number is not valid.

The program is in two parts, to make customisation or integration in another routine into your own program easier. The first part, lines 180 to 190, inputs the VAT number, sends it to the subroutine that actually checks the number, and then interprets and prints the result. It then asks whether the user wishes to check another number. If they do, the process is repeated. If not, the program ENDS on line 180. The second part of the program, lines 200-260, contains the checking routine itself.

The checking routine retrieves the result of its check in two variables V and E. If V=1, the VAT number is not valid. If V=1, the VAT number is valid. However, before the validity of the number is checked, the input itself is checked for errors. If E is not equal to zero upon return from the routine, it means that such an error has been detected. If E=1 the input contains less than nine characters (the length of any VAT number). If E=2 the input contains more than nine characters. If E=3 the input contains non-numeric characters. If any of these errors are detected, V is always set to zero, since the input could not have been a genuine VAT number. It should be

noted that spaces are stripped from the input before it is checked. This usefully enables VAT numbers to be input inclusive of spaces, as they so often appear on invoices. Upon return from the subroutine, V3 will contain the original input, stripped of spaces.

The whole program can be run as a single subroutine by simply replacing END in line 180 with RETURN. However, should you wish to use only the checking subroutine (lines 200-260), then the VAT number to be checked must first be put into AB before executing a GO SUB to the routine. As we have seen, the result of the check will be returned in variables V and E. Since there are no GOTOs within the checking subroutine, recombining it for incorporation into one's own programs is simplicity itself.

The calculation used in this program has been extensively tested, and is a form of the calculation used by the Tax Office to create the VAT numbers in the first place. For obvious reasons, it was thought rather counter productive to include that calculation in this program. As an incidental observation, it would appear that using their present system, it is only possible for the Tax Office to issue a maximum of one million VAT numbers.



PROGRAM: VAT CHECKER

```

20 100 REM *** VAT NUMBER CHECKER ***
21 110 ESC120=“100 PGM”, ESC20=“T
22 00 PGM”; ESC130=“NON-NUMERIC”
23 180 RS=“”; INPUT“ENTER VAT NU
24 MBER”; RS: IF RS=“” THEN PRINT“NO
25 INPUT”; GO TO 180
26 190 GO SUB 200: IF E<>0 THEN PRINT“
27 ERROR: RS IS ”; RS:“”; CH
28 ARACTERS”
29 190 PRINT“VAT NUMBER IS “; IF E
30 TSC=“NOT ”, ABS(V-1)=0 THEN PRINT“
31 0”
32 180 PRINT“TRY AGAIN (Y/N) ? ”
33 190 GET DS: IF DS=“Y” AND DS<>“N
34 ” THEN 190
35 170 PRINT DS: IF DS=“Y” THEN 120
36 180 END
37 200 REM *** CHECK ROUTINE ***
38 210 US=“”; FORM=1; TOL=1E-03; US
39 =US+RIDS(CS,A, RSKH(CS,C),A,
40 13)*“”; T3: NEXT
41 220 U=0; C=0; B=0; IF LEN(CS)<>9
42 THEN=RSCLN(DS); S3=1: RETURN
43 230 FORM=1; DS: IF A<T3 THEN B=0+U
44 RIDS(CS,A, 13)*C-A
45 240 IF RIDS(A, A, 13)*0>RIDS
46 (A, A, 13)*“0” THEN=0; A=0
47 250 NEXT: IF E<>0 THEN RETURN
48 260 V=RS(CS-(INT(B/97)*1)*77
49 )=VAL(CRHTS(CS, 23)): RETURN

```

MOVING THE SCREEN

by Ewan Villiers

There are many great myths in the world such as Atlantis, U.F.O.s and, probably the greatest of them all, the difficulty of moving the text and character memory round. This program has been written to end the last of those myths.

Moving the text screen has many uses, take for example menus in adventures and windows in word processors. These need not have the text beneath them and one method of doing this is to move the screen under interrupt (if the menu isn't over the text it can't harm it). This program will also allow you to make small alterations to the character set without needing to use a character designer.

I have included two versions of this program in the listings. The first is a machine code version (for speed) and the second a Basic listing (for ease of understanding). All lazy readers should note that the Basic listing can be typed in and compiled to save missing around with data statements.

The programme requires 3 variables:-

1. A Complex Interface Adapter Bank (C.I.A.)
2. A screen pointer within the bank.
3. A character pointer

THE C.I.A. BANK:

The chip which looks at the memory can only 'see' 16K at one time so this sets the block to be 'seen'.

SCREEN POINTER:

The C.I.A. bank picks out a block of 16K. The screen pointer cuts this into 1K blocks which mark the start of each screen.

CHARACTER POINTER:

This is like the screen pointer except that, as the primary character set is 16K long, it is cut into 2K blocks (I haven't copied the second set so try changing it when you run the program).

This is all you need to know to run the program (the addresses of the parameters are in the listing). But if you wish to understand the program, read on.

Changing the C.I.A. BANK (Lines 270-290)

The first thing to note about this is that the bank numbers run in the wrong order. The number 0 denotes the fourth block (49132-65535) and the number 3 denotes the first block (0-16384). This is sorted on line 80 by subtracting three from the bank number.

It should also be noted that before the bank can be changed, the C.I.A. must be set for input. This is done by ORing 56078 (and 152) with the

bank number (0-3). The only thing left to do is to inform the ROM that you have moved the screen (it is another bank now). The location 648 holds the screen address divided by 256. To inform the ROM, you need to OR 648 (and 8) with 54² the video bank. Once this is done the bank is changed and you can change the screen address.

Changing the Screen Address (Lines 340-350)

This is easier than changing the C.I.A. bank. The upper 4 bits of address 23222 hold the screen pointer so you just OR 53272 (and 15) with 16² the screen pointer. Again, the ROM must be informed of the screen's movement. This is accomplished by ORing 648 (and 182) with 4² the screen pointer. The screen has now been moved and you can now move the character set.

Changing the character address (Line 400)

This is the easiest part of the program. The lower 4 bits of 53272 hold the character pointer. As the character pointer runs in 2K blocks, twice the pointer is stored there. This is done by ORing 53272 (and 241) with 2² the character pointer.

The C.I.A. screen and character have now been moved but you will run into this programme because you have not copied the character set yet.

Copying the character data (lines 100-230)

While this is not difficult, it is the most difficult part of this program.

The difficulty is in looking at the character ROM and making sure you are not interrupted. These problems are solved by two passes, one to address 58334, to ensure that you are not interrupted, and the other to address 1 to allow you to look at the character ROM. The rest of this part of the program just copies memory and sets addresses 1 and 58334 back to their original values.

故 事

Doing this in Basic is much easier for 128 users. The process of copying the character ROM only requires you to use the command Bank 14 to gain access to the character ROM, but a full guide to this can be found on pages 260-263 of the 128 reference guide.

The Machine code program

This is a short program encoded in data statements which has only 2 major differences from the Basic version.

The first difference is that while in the basic program numbers are entered from 1 to n, in the machine code version they must be in the form 0 to 9 so the program will not search

The second difference is in the order of the subroutines. In this program, the main moving routine is last, so it can be called upon independently by an S73 command (S73000000).

The position of the parameters (addresses 40132-54) is shown in the listing, as are the SVS addresses, so all I can say now is "happy occurrences".

SPECIAL

Produce those colour
fades with this simple
to use program

By R. Hoben

If you have ever looked that little bit closer at a game of software, you will notice many little touches that add a little bit more to the presentation. These touches can be anything from swivelling sprites to colour effects. In this article I would like to discuss the use of colour to produce a fade effect, which when used in conjunction with your own program can create great life and interaction screens.

The first thing we must define is what exactly is a fade? A fade is basically where something, be it a sprite, gradually appears or disappears into the background. When an item appears, it is known as a fade-in and when it disappears, a fade-out.

In order to make the item fade, we must cycle it through a sequence of colours. To help you understand this, we will take a practical example. I want an instruction screen faded in from a black background and the instruction to end up in the colour yellow. We must now create the necessary colour sequence. The first colour should be the background colour - which in this case is black (000). We now want the text to gradually move from dark to light. The colours best suited for this are the three grey colours. So we add: Dark grey (191), Medium grey (99) and light grey (50).

Next we have an intermediate colour, which in this case is white (100). This intermediate colour is

preferably the opposite of the background. Lastly we put our final colour in which is yellow (207). If we had chosen the final colour to be light red then we could have put dark red in after white and then light red in order to provide a smoother colour transition. If you look at diagram 1 then you will see this whole example simplified.

Hopefully you should have followed all that. If you look at diagram 2, you will see the complete sequence for our example. So how do you use this sequence in your own programs?

If you type `..FADER`, `LOAD4K`, which is in Basic, and run it, it will generate a program on disk called `FADER.MC`. Now in your Basic or M/C program all you have

PAGE THEORY: DIAGRAM 1



BEFORE VISUAL,
THE LETTER IS THE
SAME COLOUR AS
THE BACKGROUND.

THE
INTERMEDIATE
COLOUR

FINAL
COLOUR

Берілгендердегі ғылыми

COLOUR	BLACK	CHERRY	MIDNIGHT	TOPAZ	WHITE	YELLOW
DECIMAL	0	11	12	18	1	2
HEX	#00	#99	#CC	#FF	#00	#FF

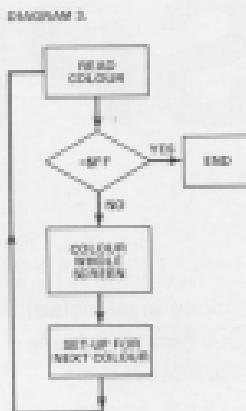
to do is load PAPER MC and then
poke the values for pixel sequence
into location \$C0015 (49205) onwards,
terminating the sequence with EFT
(\$00). Now all you have to do is print
the text you want to be faded onto
the screen at the SAME colour as
the background. E.g. if you are
fading the text in from a black
background, then print the text onto
the screen in black. It is then only
a matter of typing CN49151 and key
press!

If you want to see this in action, type in FADER.DEMO and run it. Note, you must have FADER.MC on the tape or disk first. I have presented the demo to do the fades in as in the example I have just given. Now that

you know how to fade-in, you can fade-out simply by reversing the sequence.

This fade technique can be applied to sprites and high resolution screens. Hopefully you should be able to develop your own programs to do this. An example of doing a fade with sprites is given in the program SPRITE FADE.

On a final note, you should have no problems typing in the programs as they are all in Basic. Also for those of you interested, I have given a rough flow diagram for PADDIE, which is shown in diagram 3. So there you have everything you need and hopefully you should be able to put this excellent technique to good use.



REFERENCES

TAPESTRERS WILL HAVE TO ALTER ALL THE 3646 JS.



卷之三

ANSWER

• a **recomendado** para quem, de
modo geral, é

Strung Along

Understand the use of strings with this clear, concise explanation

By E. Dumbill

It is common knowledge that Commodore Basic is far from easy to use at the best of times. This problem may have put many people off trying to program for themselves. One of the main inadequacies in the language is the functions designed to handle string variables, that is, LEFT\$, RIGHTS and MID\$. I will discuss the use of these functions in detail later. Basically, they allow the programmer to extract 'chunks' from string variables to manipulate within the program. Very well, you say, so what is wrong with that? Well, the problem is that while you can use LEFT\$, etc, to extract bits from strings, see figure 1, you are not allowed to surgically change parts of a string, see figure 2.

Figure 1

```
AS="ABC"
PRINT LEFT$(AS,2)
the result would be: AB
```

Figure 2

AS="ABC"

Now, supposing you wanted to change the 'AB' to 'DE' you might type:

LEFT\$(AS,2)+"DE"

the result would be: ~~Popatas~~ area

As you can see, the computer will let you see what is there, but you are not permitted to change it. Many other forms of Basic do allow this function. Wouldn't it be nice if, just like using DEF FN in arithmetic, we could define functions to allow us to alter strings? But, you have guessed it, C64 Basic will not allow it. I thought that this was inconvenient to say the least, and so I produced a series of subroutines which would allow me to alter my strings.

I will start at the beginning and describe to you how each function works, and how the surgical version of it works.

LEFT\$

This function lets you take the leftmost characters from a string. The syntax is: LEFT\$(string, elements), where elements = the amount of characters you wish to extract. This number can obviously not be greater than the number of characters in the string.

RIGHTS

This function is very similar to LEFTS, in that it takes the rightmost characters from the string. The syntax is identical to that of LEFTS.

MIDS

This function allows you to take characters from the MIDDLE of a string, starting at the character that you specify. It is a very useful function and has many applications. For example, searching through a string to find a key character (perhaps 'C'). The syntax is: MIDS (string\$, start, elements) where start is the start character and elements is the amount of characters you wish to extract. Figure 1 shows a very simple example.

Figure 1

```
AS="ABCDE"
PRINT MIDS(AS,2,2)
the result would be: C'D
```

Now you know how to use the main functions, we can use these functions to help us define routines to alter strings as our lesson. Let's start with LEFTS. If we are altering the LEFTS part of a string, we are obviously leaving away the RIGHTS part of the string. With this in mind, we can say that:

```
altered STRING$=NEW BITS+RIGHTS (OLD STRING$)
```

But, we do not know how many characters to change in the LEFTS, and consequently, how many RIGHTS characters (represented by \$) to put on the end of the altered string to make it complete. For this, we need to use another function LEN. This returns the number of characters in a string, referring to Figure 1 the information:

```
PRINT LEN(AS)
would result in 5
```

If we can call the number of characters we want to leave intact is LEN (STRIPED) times N. We can put this into our program using LEFTS syntax.

```
altered STRING$=NEW BITS+RIGHTS (OLD STRING$*N)
```

But, it is a pain to have to state both the NEW BITS, so we obviously need to use our friend LEN, and define N as LEN (NEW BITS). So, our revised program looks like this:

```
ALTERED: STRING$=NEW BITS+RIGHTS (OLD STRING$, LEN (OLD STRING$)-LEN (NEW BITS))
```

Complicated isn't it? No, not really! Here is a breakdown of the ugly looking RIGHTS function I used. What it does is to take the original string, and to chop off the LEFTS that is to be replaced by NEW BITS. The length of the bit remaining intact (the RIGHTS) is given by subtracting the length of the replaced bit (NEW BITS) from the length of the original string.

Whew!

Now that we have struggled through that, we can actually do something with it, and write a routine to use in our programs.

Here is a sample example of the routine and how to call it:

```
100 LS=RIGHT$(LS,LEN(LS)-LEN(NS))
110 RETURN
200 LS="CAT SAT ON THE MAT"
210 NS="DAN": GOSUB100
```

If you now run with GOTO200, then type PRINT LS, boy pastor!, the using LS will have changed to DAN SAT ON THE MAT. Just a few notes about the routine: You will always have to use LS and NS for the strings to be specified, as there is no way of creating a new function. What we are doing is making a program that uses global variables, and there is no way of making values of variables stay ONLY inside the routine (LOCAL) as there is in other basics. That is the major limitation of the routine. So if, for example, you are using the variable WS, and wished to alter it, you would need to make LS=WS before calling the routine, and WS=LS after calling the routine. For this reason, you may find it more convenient to include line 100 as part of the main program, instead of using it as part of the subroutine. This too has its disadvantages, such as getting a sore finger from repeatedly typing brackets!

The subroutine for a surgical RIGHTS is obviously going to be much the same, but we must exchange the RIGHTS in the routine for a LEFT\$, as it will be the LEFT\$ that we will now want to have altered!

Here is the routine for a surgical RIGHTS:

```
100 LS=LEFT$(LS,LEN(LS)-LEN(NS))-NS
110 RETURN
```

The use of this routine is much the same as the one above. How simple you say! Doesn't it look easy? Well, you'd dare, but then we discover a stumbling block in the form of MIDS. However, with a bit of logical thought we can overcome it.

Just as a string is composed of LEFT\$ and RIGHTS, it is also composed of LEFT\$, MIDS and RIGHTS. Therefore, we now get:

ALTERED STRINGS=LEFT\$(OLDS\$)+NEW MID\$+RIGHT\$(OLDS\$)

As with MIDS, we can't get away with 2 parameters, NS and LS, but we need a third, S, which is the character at which the MID\$ will start being inserted.

From that S, we can calculate all the information that we may need:

number of characters in LEFT\$+S
number of characters in RIGHTS+LEN(LS)-S-LEN(NS)
so the program for MIDS surgery looks like this:

```
100
LS=LEFT$(LS,LEN(LS)-NS-RIGHT$(LS,LEN(LS)-S-LEN(NS)))
110 RETURN
```

Use is as for the routines using LEFT\$ and RIGHTS, but with the extra inclusion of S, the start character for the insertion of NS.

I hope that you have enjoyed and understood this approach to constructing program routines to the end of making your own functions. Also, I hope that you can appreciate the power that a little thought adds to the humble Commodore.

Keep stringing!

EXTENDING BASIC 8 PART

Add an auto line
numbering facility to
your collection of
extended Basic routines

By Burghard-Henry
Lehmann

When you enter a Basic textile, the one thing which is fairly regular is the line numbers. Since computers are very good at doing regular, monotonous tasks, it makes sense to let the computer do the line numbering. All you have to do then, is worry about designing your program.

Automatic line numbering is pretty easy. All we have to do is intercept the flow of Basic in the ROM after a line has been entered into the textile, or before a new line is started off - whichever way you want to look at it. For this let me elaborate on how Commodore Basic deals with a newly entered line.

Basic Warm Start

The whole process starts at \$A483, the so called Basic Warm Start routine. This is the central point to which Basic loops back each time a line has been entered into the textile or a direct command has been executed.

This point is so important, that it has been reviewed by those clever Commodore ROM designers. That is, instead of jumping straight to \$A483, the computer fetches the address it has to jump to from the vector at \$0002. Under normal circumstances this location which lies in RAM and can be changed by the programmer, contains \$A483 - the Basic Warm Start routine.

This will be the point where we will intercept Basic to introduce our auto line numbering routine. More about this in a minute.

At the beginning of the Warm Start routine, the computer goes into a loop which waits for the user to enter a character on the keyboard. This character, which can be anything at this point, is stored in a location, called the input buffer (\$0000). Next the computer waits for the user to enter another character, this is stored in the following location in the input buffer, and so on until the user presses the return key (ASCII 13). This finishes this loop and terminates the character in the input buffer with a zero.

Now, the computer thinks out, if the line entered has a line number in front of it or not. If it hasn't got a line number, the "statement", as it is called, is interpreted and executed immediately as a direct command.

If the line starts with a line number, the keywords in the line are converted into tokens and then the computer looks if a line with the same line number is already present in the textile.

If a line with that line number exists already, the old line is deleted.

Finally, the new line is inserted into the textile.

If the line has nothing after the line number, nothing is inserted into the textile, thus if this line already exists it will be deleted.

Auto Line Numbering

To introduce auto-line numbering we first need a new command which starts it off.

I have chosen to use the standard extended command AUTO. Note, when testing for AUTO, the last two letters of the command will be tokenised by the computer into \$A4, since TO is a Commodore Basic function.

To produce line numbers automatically we need to know two things:
Which line number does the user want to start automatic line numbering?
What step or increment does he want the line numbers to use.

AUTO therefore needs two parameters:

The line number to start with.

The increment up to the next line number.

Both parameters should be separated with a comma. I haven't bothered to build in any error checks. If you want to be perfect, you should know by now how to fiddle around something like this and force the computer to give a syntax error report.

The auto routine itself starts by collecting these two parameters and saving them into the zero page locations 251/252 (first line number) and 253/254 (step) (lines 1646-1781).

Next we change the Warm Start vector so that it points at our routine, instead of the usual routine in the ROM (lines 1636-1830).

Now we print a carriage return, which puts the cursor onto the beginning of the next line (lines 1890-1898).

Now we are all set to do auto line numbering.

The overall mechanics of auto line numbering is as follows. The current line number, contained in 251/252 is printed on the screen (lines 1990-2040). Then it is copied into the beginning of the input buffer (lines 2080-2150).

To print the number on screen we use the normal ROM routine \$8D\$CD, which prints any number whose low byte is in X and whose high byte is in the accumulator. So that the routine can print the number onto the screen, it naturally has to convert it into ASCII digits. It does this and stores the ASCII digits at location 10400, terminated with a zero.

The reason why I explain all this is, that we make use of this fact when we have to move the line number (in ASCII digits) into the input buffer (lines 2080-2130).

Now the computer enters the usual loop in ROM, which I have explained above, and which waits for the user to enter a full line (line 2170). In this routine the X-register is used for the index to the input buffer. Normally the loop starts with X=0, but since we have already entered a line number into the input buffer, X contains whatever number of digits the line number consists of. We have the correct amount in X from the loop which we used earlier on to move the ASCII digits from location 10400 into the input buffer.

After this, we test for direct or line numbered statements (lines 2210-2280). This is a little capricious here, since there is no question about a direct statement being given at this point.

Line 2260 calls a routine in ROM which converts the line number at the beginning of the input buffer from ASCII into the more useful low byte/high byte sequence and stores the result (see page 154/15). This is used later to search the textfile to see if the line number already exists.

Now all basic keywords in the line are tokenised (line 2440).

Then the computer searches through the textfile to see if a line with this number already exists (line 2480). If it does, the line is deleted (line 2580).

In lines 2580-2680 we calculate the next line number by adding the step contained in 253/254 to the previous line number contained in 251/252.

Then there are two further ROM routines (lines 2680 and 2690) which re-organise the basic textfile and reset all the necessary pointers.

In lines 2710-2730 comes an important test which sees if there is something after the line number.

Most of the previous code I have copied straight from the ROM. Why this seemingly unnecessary duplication of code, you may ask?

Well, as I have already pointed out, it would be pretty annoying if we couldn't switch auto line numbering off once while auto line numbering is on you can't give a direct command. Furthermore, the vectors at \$0380 to \$0390

```

1636 0000 0000
1637
1638 0000 0000
1639 0000 0000
1640 0000 0000
1641 0000 0000
1642 0000 0000
1643 0000 0000
1644 0000 0000
1645 0000 0000
1646 0000 0000
1647 0000 0000
1648 0000 0000
1649 0000 0000
1650 0000 0000
1651 0000 0000
1652 0000 0000
1653 0000 0000
1654 0000 0000
1655 0000 0000
1656 0000 0000
1657 0000 0000
1658 0000 0000
1659 0000 0000
1660 0000 0000
1661 0000 0000
1662 0000 0000
1663 0000 0000
1664 0000 0000
1665 0000 0000
1666 0000 0000
1667 0000 0000
1668 0000 0000
1669 0000 0000
1670 0000 0000
1671 0000 0000
1672 0000 0000
1673 0000 0000
1674 0000 0000
1675 0000 0000
1676 0000 0000
1677 0000 0000
1678 0000 0000
1679 0000 0000
1680 0000 0000
1681 0000 0000
1682 0000 0000
1683 0000 0000
1684 0000 0000
1685 0000 0000
1686 0000 0000
1687 0000 0000
1688 0000 0000
1689 0000 0000
1690 0000 0000
1691 0000 0000
1692 0000 0000
1693 0000 0000
1694 0000 0000
1695 0000 0000
1696 0000 0000
1697 0000 0000
1698 0000 0000
1699 0000 0000
1700 0000 0000
1701 0000 0000
1702 0000 0000
1703 0000 0000
1704 0000 0000
1705 0000 0000
1706 0000 0000
1707 0000 0000
1708 0000 0000
1709 0000 0000
1710 0000 0000
1711 0000 0000
1712 0000 0000
1713 0000 0000
1714 0000 0000
1715 0000 0000
1716 0000 0000
1717 0000 0000
1718 0000 0000
1719 0000 0000
1720 0000 0000
1721 0000 0000
1722 0000 0000
1723 0000 0000
1724 0000 0000
1725 0000 0000
1726 0000 0000
1727 0000 0000
1728 0000 0000
1729 0000 0000
1730 0000 0000
1731 0000 0000
1732 0000 0000
1733 0000 0000
1734 0000 0000
1735 0000 0000
1736 0000 0000
1737 0000 0000
1738 0000 0000
1739 0000 0000
1740 0000 0000
1741 0000 0000
1742 0000 0000
1743 0000 0000
1744 0000 0000
1745 0000 0000
1746 0000 0000
1747 0000 0000
1748 0000 0000
1749 0000 0000
1750 0000 0000
1751 0000 0000
1752 0000 0000
1753 0000 0000
1754 0000 0000
1755 0000 0000
1756 0000 0000
1757 0000 0000
1758 0000 0000
1759 0000 0000
1760 0000 0000
1761 0000 0000
1762 0000 0000
1763 0000 0000
1764 0000 0000
1765 0000 0000
1766 0000 0000
1767 0000 0000
1768 0000 0000
1769 0000 0000
1770 0000 0000
1771 0000 0000
1772 0000 0000
1773 0000 0000
1774 0000 0000
1775 0000 0000
1776 0000 0000
1777 0000 0000
1778 0000 0000
1779 0000 0000
1780 0000 0000
1781 0000 0000
1782 0000 0000
1783 0000 0000
1784 0000 0000
1785 0000 0000
1786 0000 0000
1787 0000 0000
1788 0000 0000
1789 0000 0000
1790 0000 0000
1791 0000 0000
1792 0000 0000
1793 0000 0000
1794 0000 0000
1795 0000 0000
1796 0000 0000
1797 0000 0000
1798 0000 0000
1799 0000 0000
1800 0000 0000
1801 0000 0000
1802 0000 0000
1803 0000 0000
1804 0000 0000
1805 0000 0000
1806 0000 0000
1807 0000 0000
1808 0000 0000
1809 0000 0000
1810 0000 0000
1811 0000 0000
1812 0000 0000
1813 0000 0000
1814 0000 0000
1815 0000 0000
1816 0000 0000
1817 0000 0000
1818 0000 0000
1819 0000 0000
1820 0000 0000
1821 0000 0000
1822 0000 0000
1823 0000 0000
1824 0000 0000
1825 0000 0000
1826 0000 0000
1827 0000 0000
1828 0000 0000
1829 0000 0000
1830 0000 0000
1831 0000 0000
1832 0000 0000
1833 0000 0000
1834 0000 0000
1835 0000 0000
1836 0000 0000
1837 0000 0000
1838 0000 0000
1839 0000 0000
1840 0000 0000
1841 0000 0000
1842 0000 0000
1843 0000 0000
1844 0000 0000
1845 0000 0000
1846 0000 0000
1847 0000 0000
1848 0000 0000
1849 0000 0000
1850 0000 0000
1851 0000 0000
1852 0000 0000
1853 0000 0000
1854 0000 0000
1855 0000 0000
1856 0000 0000
1857 0000 0000
1858 0000 0000
1859 0000 0000
1860 0000 0000
1861 0000 0000
1862 0000 0000
1863 0000 0000
1864 0000 0000
1865 0000 0000
1866 0000 0000
1867 0000 0000
1868 0000 0000
1869 0000 0000
1870 0000 0000
1871 0000 0000
1872 0000 0000
1873 0000 0000
1874 0000 0000
1875 0000 0000
1876 0000 0000
1877 0000 0000
1878 0000 0000
1879 0000 0000
1880 0000 0000
1881 0000 0000
1882 0000 0000
1883 0000 0000
1884 0000 0000
1885 0000 0000
1886 0000 0000
1887 0000 0000
1888 0000 0000
1889 0000 0000
1890 0000 0000
1891 0000 0000
1892 0000 0000
1893 0000 0000
1894 0000 0000
1895 0000 0000
1896 0000 0000
1897 0000 0000
1898 0000 0000
1899 0000 0000
1900 0000 0000
1901 0000 0000
1902 0000 0000
1903 0000 0000
1904 0000 0000
1905 0000 0000
1906 0000 0000
1907 0000 0000
1908 0000 0000
1909 0000 0000
1910 0000 0000
1911 0000 0000
1912 0000 0000
1913 0000 0000
1914 0000 0000
1915 0000 0000
1916 0000 0000
1917 0000 0000
1918 0000 0000
1919 0000 0000
1920 0000 0000
1921 0000 0000
1922 0000 0000
1923 0000 0000
1924 0000 0000
1925 0000 0000
1926 0000 0000
1927 0000 0000
1928 0000 0000
1929 0000 0000
1930 0000 0000
1931 0000 0000
1932 0000 0000
1933 0000 0000
1934 0000 0000
1935 0000 0000
1936 0000 0000
1937 0000 0000
1938 0000 0000
1939 0000 0000
1940 0000 0000
1941 0000 0000
1942 0000 0000
1943 0000 0000
1944 0000 0000
1945 0000 0000
1946 0000 0000
1947 0000 0000
1948 0000 0000
1949 0000 0000
1950 0000 0000
1951 0000 0000
1952 0000 0000
1953 0000 0000
1954 0000 0000
1955 0000 0000
1956 0000 0000
1957 0000 0000
1958 0000 0000
1959 0000 0000
1960 0000 0000
1961 0000 0000
1962 0000 0000
1963 0000 0000
1964 0000 0000
1965 0000 0000
1966 0000 0000
1967 0000 0000
1968 0000 0000
1969 0000 0000
1970 0000 0000
1971 0000 0000
1972 0000 0000
1973 0000 0000
1974 0000 0000
1975 0000 0000
1976 0000 0000
1977 0000 0000
1978 0000 0000
1979 0000 0000
1980 0000 0000
1981 0000 0000
1982 0000 0000
1983 0000 0000
1984 0000 0000
1985 0000 0000
1986 0000 0000
1987 0000 0000
1988 0000 0000
1989 0000 0000
1990 0000 0000
1991 0000 0000
1992 0000 0000
1993 0000 0000
1994 0000 0000
1995 0000 0000
1996 0000 0000
1997 0000 0000
1998 0000 0000
1999 0000 0000
2000 0000 0000
2001 0000 0000
2002 0000 0000
2003 0000 0000
2004 0000 0000
2005 0000 0000
2006 0000 0000
2007 0000 0000
2008 0000 0000
2009 0000 0000
2010 0000 0000
2011 0000 0000
2012 0000 0000
2013 0000 0000
2014 0000 0000
2015 0000 0000
2016 0000 0000
2017 0000 0000
2018 0000 0000
2019 0000 0000
2020 0000 0000
2021 0000 0000
2022 0000 0000
2023 0000 0000
2024 0000 0000
2025 0000 0000
2026 0000 0000
2027 0000 0000
2028 0000 0000
2029 0000 0000
2030 0000 0000
2031 0000 0000
2032 0000 0000
2033 0000 0000
2034 0000 0000
2035 0000 0000
2036 0000 0000
2037 0000 0000
2038 0000 0000
2039 0000 0000
2040 0000 0000
2041 0000 0000
2042 0000 0000
2043 0000 0000
2044 0000 0000
2045 0000 0000
2046 0000 0000
2047 0000 0000
2048 0000 0000
2049 0000 0000
2050 0000 0000
2051 0000 0000
2052 0000 0000
2053 0000 0000
2054 0000 0000
2055 0000 0000
2056 0000 0000
2057 0000 0000
2058 0000 0000
2059 0000 0000
2060 0000 0000
2061 0000 0000
2062 0000 0000
2063 0000 0000
2064 0000 0000
2065 0000 0000
2066 0000 0000
2067 0000 0000
2068 0000 0000
2069 0000 0000
2070 0000 0000
2071 0000 0000
2072 0000 0000
2073 0000 0000
2074 0000 0000
2075 0000 0000
2076 0000 0000
2077 0000 0000
2078 0000 0000
2079 0000 0000
2080 0000 0000
2081 0000 0000
2082 0000 0000
2083 0000 0000
2084 0000 0000
2085 0000 0000
2086 0000 0000
2087 0000 0000
2088 0000 0000
2089 0000 0000
2090 0000 0000
2091 0000 0000
2092 0000 0000
2093 0000 0000
2094 0000 0000
2095 0000 0000
2096 0000 0000
2097 0000 0000
2098 0000 0000
2099 0000 0000
2100 0000 0000
2101 0000 0000
2102 0000 0000
2103 0000 0000
2104 0000 0000
2105 0000 0000
2106 0000 0000
2107 0000 0000
2108 0000 0000
2109 0000 0000
2110 0000 0000
2111 0000 0000
2112 0000 0000
2113 0000 0000
2114 0000 0000
2115 0000 0000
2116 0000 0000
2117 0000 0000
2118 0000 0000
2119 0000 0000
2120 0000 0000
2121 0000 0000
2122 0000 0000
2123 0000 0000
2124 0000 0000
2125 0000 0000
2126 0000 0000
2127 0000 0000
2128 0000 0000
2129 0000 0000
2130 0000 0000
2131 0000 0000
2132 0000 0000
2133 0000 0000
2134 0000 0000
2135 0000 0000
2136 0000 0000
2137 0000 0000
2138 0000 0000
2139 0000 0000
2140 0000 0000
2141 0000 0000
2142 0000 0000
2143 0000 0000
2144 0000 0000
2145 0000 0000
2146 0000 0000
2147 0000 0000
2148 0000 0000
2149 0000 0000
2150 0000 0000
2151 0000 0000
2152 0000 0000
2153 0000 0000
2154 0000 0000
2155 0000 0000
2156 0000 0000
2157 0000 0000
2158 0000 0000
2159 0000 0000
2160 0000 0000
2161 0000 0000
2162 0000 0000
2163 0000 0000
2164 0000 0000
2165 0000 0000
2166 0000 0000
2167 0000 0000
2168 0000 0000
2169 0000 0000
2170 0000 0000
2171 0000 0000
2172 0000 0000
2173 0000 0000
2174 0000 0000
2175 0000 0000
2176 0000 0000
2177 0000 0000
2178 0000 0000
2179 0000 0000
2180 0000 0000
2181 0000 0000
2182 0000 0000
2183 0000 0000
2184 0000 0000
2185 0000 0000
2186 0000 0000
2187 0000 0000
2188 0000 0000
2189 0000 0000
2190 0000 0000
2191 0000 0000
2192 0000 0000
2193 0000 0000
2194 0000 0000
2195 0000 0000
2196 0000 0000
2197 0000 0000
2198 0000 0000
2199 0000 0000
2200 0000 0000
2201 0000 0000
2202 0000 0000
2203 0000 0000
2204 0000 0000
2205 0000 0000
2206 0000 0000
2207 0000 0000
2208 0000 0000
2209 0000 0000
2210 0000 0000
2211 0000 0000
2212 0000 0000
2213 0000 0000
2214 0000 0000
2215 0000 0000
2216 0000 0000
2217 0000 0000
2218 0000 0000
2219 0000 0000
2220 0000 0000
2221 0000 0000
2222 0000 0000
2223 0000 0000
2224 0000 0000
2225 0000 0000
2226 0000 0000
2227 0000 0000
2228 0000 0000
2229 0000 0000
2230 0000 0000
2231 0000 0000
2232 0000 0000
2233 0000 0000
2234 0000 0000
2235 0000 0000
2236 0000 0000
2237 0000 0000
2238 0000 0000
2239 0000 0000
2240 0000 0000
2241 0000 0000
2242 0000 0000
2243 0000 0000
2244 0000 0000
2245 0000 0000
2246 0000 0000
2247 0000 0000
2248 0000 0000
2249 0000 0000
2250 0000 0000
2251 0000 0000
2252 0000 0000
2253 0000 0000
2254 0000 0000
2255 0000 0000
2256 0000 0000
2257 0000 0000
2258 0000 0000
2259 0000 0000
2260 0000 0000
2261 0000 0000
2262 0000 0000
2263 0000 0000
2264 0000 0000
2265 0000 0000
2266 0000 0000
2267 0000 0000
2268 0000 0000
2269 0000 0000
2270 0000 0000
2271 0000 0000
2272 0000 0000
2273 0000 0000
2274 0000 0000
2275 0000 0000
2276 0000 0000
2277 0000 0000
2278 0000 0000
2279 0000 0000
2280 0000 0000
2281 0000 0000
2282 0000 0000
2283 0000 0000
2284 0000 0000
2285 0000 0000
2286 0000 0000
2287 0000 0000
2288 0000 0000
2289 0000 0000
2290 0000 0000
2291 0000 0000
2292 0000 0000
2293 0000 0000
2294 0000 0000
2295 0000 0000
2296 0000 0000
2297 0000 0000
2298 0000 0000
2299 0000 0000
2300 0000 0000
2301 0000 0000
2302 0000 0000
2303 0000 0000
2304 0000 0000
2305 0000 0000
2306 0000 0000
2307 0000 0000
2308 0000 0000
2309 0000 0000
2310 0000 0000
2311 0000 0000
2312 0000 0000
2313 0000 0000
2314 0000 0000
2315 0000 0000
2316 0000 0000
2317 0000 0000
2318 0000 0000
2319 0000 0000
2320 0000 0000
2321 0000 0000
2322 0000 0000
2323 0000 0000
2324 0000 0000
2325 0000 0000
2326 0000 0000
2327 0000 0000
2328 0000 0000
2329 0000 0000
2330 0000 0000
2331 0000 0000
2332 0000 0000
2333 0000 0000
2334 0000 0000
2335 0000 0000
2336 0000 0000
2337 
```

are not charged back to normal, even when you see the sun/stop radios level So, we have to take care of this ourselves.

It makes sense to switch auto line numbering off, when the user enters numbers for the line number.

In this case our routine jumps to the aptly called routine `ROUTINING`, which resets the `Warm Start` vector back to normal (lines 2810-2840) and then jumps to the normal `MAIN` routine (lines 2850-2860).

By now you should understand how you go about patching in new commands to Commander BASIC and should be ready to write some of your own. Watch out for more from me in a future installment of *President BASIC*.

PROGLOK

Stop prying eyes and
Itchy fingers with this
handy program

By Zak Beck

This program enables you, having just written your latest masterpiece, to 'lock' the computer so that, say, the kids cannot spoil all your hard work. Before leaving your computer, it simply requires that you press **Y** and enter any 8-digit password. Before the computer will return back to the language (eg. BASIC), the correct code must be entered. This idea has been used before on expensive utility cartridges - well now you have it at little if no cost!

How it all works

The program is written in machine code. Obviously, if it were written in Basic, it would not be compatible with your masterpiece. It resides below Basic, from \$1300 to \$34C1, in the application or cartridge program area. This means that Basic does not have to be moved about, but the program MAY clash with some cartridges which use this area as a workspace.

When you first use the program, it sets up function key 1 to call itself, and then returns back to Basic. When you wish to 'lock' the computer, press **Y**. The program first deactivates the restore key, prints up its title, and some prompts. It does this using a kernel call called **PRIMM1** - print message. This works as follows:

JSR prints
key 'Message goes here' D

(rest of program follows.)

Having called PRIMM1, the com-

puter will print out everything that follows until it comes across a **0** byte. Then it jumps back to your subroutine. This useful message-printer resides at \$3403, \$F8F7D.

Program, having printed its title and prompts will then ask for any 8-character code. This is implemented by subroutine **getcode**, which inputs 8 characters, storing them in buffer **buff**. Then the program stores the contents into the password storage space, **pass**.

Next, prompts are printed (label **test**), to tell the user that the computer has been locked, and that to use it he has to enter a 8 character password. Using **getcode**, an attempt is entered into the buffer **buff**, and the contents of **buff** are compared with the password **pass**. If any discrepancies are found, the computer jumps back to **test**. Otherwise, the restore key is reactivated, and the computer returns to the language (eg. BASIC).

Using the Program

Type in the program (see getting it all in). Next, type the following

SYS4881 and press return

This will initialise function key 1 to read **SY\$4881** + chd (13) as the command **KEY** will show.

Pressing **Y** in direct mode will call up the program, executing a call to address 4881. You can now enter your keyword, using any of the keys on the keyboard, including Line Feed and those on the keypad at the right of the C128. If four six characters provide a fairly secure code, who wants their computer to be Fort Knox?

The password having been

entered, the computer is immune to the restore key, and can only be accessed by typing in your code. When you come to want to use the machine again, press any key and up will come a screen asking you to enter your code. Entering the right code will take you into BASIC, entering the wrong one will take you back to the press any key screen again.

Should you forget the code, there is a useful little trick to enable you to recover your program. Follow the below instructions carefully:

1. Hold down the RUN/STOP key;
2. Press in the reset button while holding RUN/STOP;
3. When the computer powers up, you will be in the monitor. Release RUN/STOP now!
4. Type X and press return.
5. You will be back in BASIC. Typing LIST and pressing return should give you your program back.

Getting it in

The Basic loader is easy to use. Just type it all in very carefully, and save it to tape or disk. Then run it. If you've made a mistake, **TYPEING ERROR** will appear. If everything is alright, you can save the code using the following BASIC line:

SAVE "(filename)",815, P4881 TO P513

When you want to use the program next time, use

LOAD "(filename)"

SYS 4884

And then follow the instructions under 'Using the Program'.

And that's it...



www.ijerpi.com

C64 PROGRAMMING -

THE USER PORT

When Commodore created the user port, they did not intend it to be used purely for RS232 communications, but instead to form a link between the computer and the outside world.

Despite the fact that there are twenty four pins available from the user port, only ten are actually required to connect the C64 to the outside world: PWR, PRT, S, GND.

The 3v and Ground (0VDC) lines can usually be used to power anything which is connected to the C64, but you must be careful not to draw more than 100mAamps because this will damage the computer.

The theory

You should already know that there are eight lines in the user port (labeled P0-P7) which are available for the use of the individual. However, to actually use these lines they must be set to either input or output (they are set to input on power up). Luckily, the Commodore operating system makes this an extremely simple process. At location 36379 there is a bit for each line. To set a line to input, make the bit zero, to set a line to output the bit must be one, (i.e. if you set bit 3 to one then line P3 will be set to output). The following program should make this easier to understand.

BB Rec 74 34 33 16

Explore the possibilities
of this versatile user
port.

By R. Sedley

You can see that lines 6, 3 and 2 are going to be set to output, and lines 7, 5, 4, 1 and 9 set to input. To actually achieve this configuration, the binary numbers, and so the values, must be translated into decimal so that it can be entered into the computer, using the 'poke' statement. The necessary substitution is as follows:

2 is the power of the 2 in the powers of 1 + 2 in the powers of 3.

In other words 64×4 which equals 16. Therefore, to achieve the configuration in the example we would have to **POKE 16783,16**.

879

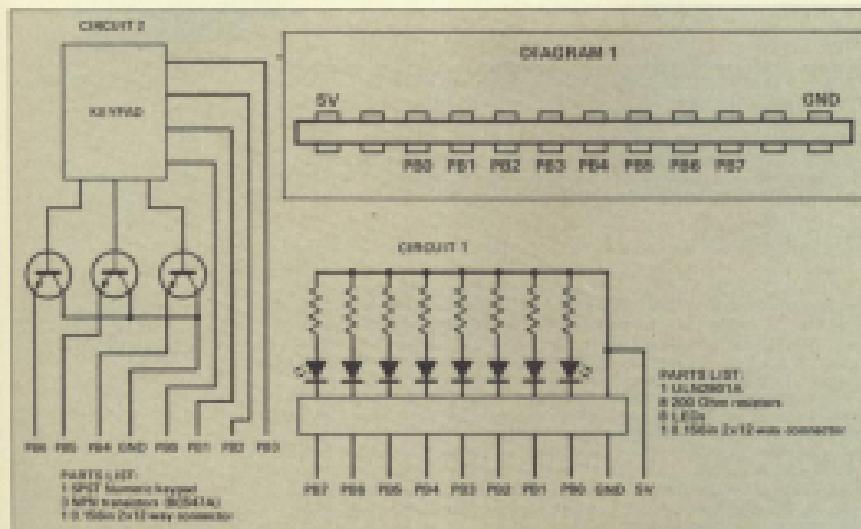
Now that the lines have been set, assuming some have been set to output, it becomes necessary to have a way of controlling the state of the lines (either low or high), which again is an extremely simple process. A bit has been allocated to each line at location \$A513. To make a line go high you set it to zero (i.e. if you set bit 6 to 1 then line PB6 will be taken to be high). You cannot however just set a bit and expect through a line set to output because this will just confuse the CPU.

The following groups should make this clear:

Bit max: 2.6.9.4.3.2.1.9
Value: 4.1.0.0.0.1.0.0

As you can see, lines 6 and 3 are required to go high, leaving the other lines low. Again, this binary figure must be translated into decimal. Using the same process as above we discover that the figure required is 68. Therefore the following statement must be entered:

PROBLEMS: LISTING 2



POTTER SMITH, 620

To enable you to see for yourselves what effect the contents of location 58277 has on the state of the lines, you might like to construct Circuit 1, (refer to Figure 7 for the pin configuration of the user port). The device contains a ULN2801A optoisolator driver which amplifies the signal from the user port, to light the LEDs on the lines which have been taken high (before using the device you must enter PDRB# 58277, 185 to set all the lines to output). In order to see the speed and accuracy at which the computer can control the lines, you might like to enter Listing 1. (As the program gains speed whilst it is running, you must remember that there is only ever 1 LED lit at a time, the reason for this being that the effect of the image lasts as long as one scan for approximately 1/60th of a second.)

1000

Assuming that some of the lines have been set to input, it becomes necessary to have a method of reading the state of the lines. As you already know, there is a bit allocated to each line at location 38577, so to read the lines all you have to do is **PEEK** this location, remembering that if some of the lines have been set to output then part of this figure will correspond to output. Because of this, and having to know which lines have been forced low by an external device, you will have to convert the original numbers from decimal to binary. The easiest way of doing this is shown below:

Bit Number: 7 6 5 4 3 2 1 0
 True Value: 128 64 32 16 8 4 2 1

To use this table to convert decimal to binary, you take your decimal number and look at the chart. Take the largest number (from the bottom row) which is smaller than your number. You then simply repeat this process until your number is reduced to zero, when you put a 1 by the bit number which did not have a 1 by them. You now have your binary figure. When you see a 1 the line is high, and a 0 shows a line which is low.

For my second example program, I thought it might be nice to come up with something which might be useful to other programmers. Consider a numeric keypad mapped to the C64. In order to see what effect pressing keys on the keypad has on the state of the lines you should consult Listing 2. To actually use the keypad, enter and run *LISTING 2* (An IRQ driven program which reads the keypad and then displays the relevant characters on the screen, because of which it will only work when the computer is in direct mode).

SKILLS OF THE JOURNEYS

The first semi-final was the Rest of the World vs The Software Publishers...

BUT THE
BALL'S DOWN
THAT'S ENDS!



2

The Rest of the World ended on 1-0
defeat!

Then it was the Journeys
match!

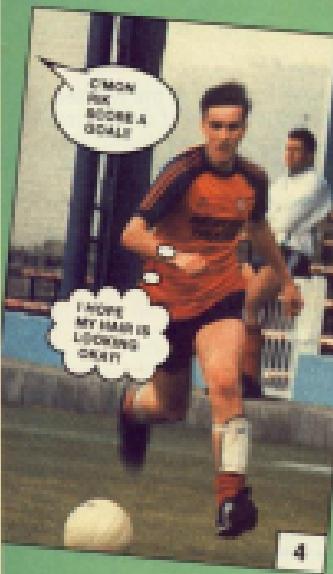
1



3

I'M ON
YOU
JOURNEYS

He didn't score, but the
Journeys obliged...



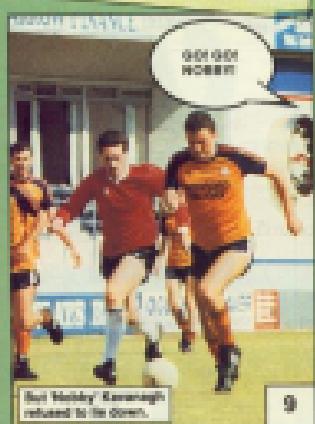
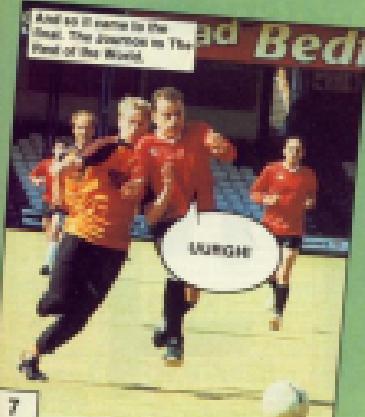
4



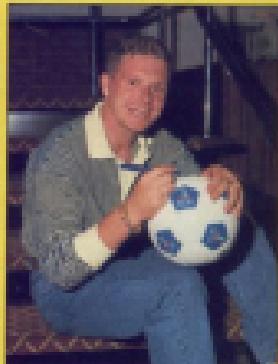
The team that they beat, The Distributors, also
got the 3rd/4th place play-off



5



Photography: Frank Mancini
(except the Anonymous)



You may ask what YC is doing to celebrate the new football season? Well we, coupled with those wobbly chappies (and chappies) at Empire, are giving away a football and a Tottenham shirt. But hold onto your aluminium studs, there are no ordinary Spurs goodies. They're Spurs goodies signed by the Man for Kid himself, Paul Gascoigne.

To get your goodies, enter on the prizes (as modified here by the delectable Maria Waddell) you must answer the following three questions. To make it harder for you we insist that you do it whilst juggling on the spot.

1. From what team did Tottenham sign Paul Gascoigne?

- a) Newcastle United.
- b) Manchester United.
- c) Washington Redskins.

2. When was the last time that Tottenham won the FA Cup?

- a) When dinosaurs did commit the Earth.
- b) 1991-92.
- c) 1990-91.

3. How tall is our Gascoigne?

- a) 2' 10".
- b) As tall as Blackpool tower.
- c) 3' 2".

Put the answers on the back of a postcard (or sealed envelope) and send them to:

Gascoigne Compo, YC, Argus House, Boundary Way, Hemel Hempstead, HP1 2SE.

The first correct card pulled out of the woolly hat on 15th November 1990 will grab the whizzies.

The Ever Important Big Wobbly Rules

Entries will not be accepted from employees of Argus Specialist Publications, or Empire. This restriction also applies to employees' families, agents of the companies, and pieces of fruit.

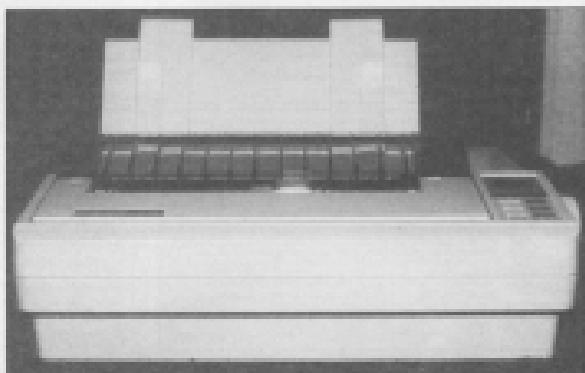
The entry restrictions form part of the rules and the Editor's decision is final (you can be guaranteed that it will be a good one though). No correspondence will be entered into. In the event of a flood, we reserve the right to use a hairdryer.

PS. No entries will be accepted if scribbled on the back of a cabinet instruction.

Q
U
E
T
S
H
I
R
T
A
Y

Could Duncan Evans be confucius? Not likely, but he has certain similarities, neither of them played full-back for Liverpool. He also runs his beades over the new Swift 24-pin printer.

SWIFT AS A NEWT



Above: A rather decorative title number, don't you think?

Confucius he say, "Computer without printer is like Steve Brett - it's all in the memory." Well I lie, he didn't really, I made that up. He would probably have had a printer though. I mean how else are you going to output your mind boggling philosophical doctrines unless you can get a hard copy. Oh, it was easy enough in Confucius' day, you just ordered the nearest monk to start scrawling on a slate and before you could say, "The analysis of Confusion", you were five generations down the line and your life's work was being published posthumously.

Unfortunately there aren't that many monks willing to spend their lives scratching out my version for *Four Commandments*, and alas, very few now either, so in order to support myself and my football team of starving children I needed the hi-tech equivalent. Citizen Europe also realised that monks were in short supply, and that they wrote as quickly as a Commodore Printer, and thus released the 120-D, a relatively cheap 9-pin printer. This combined low cost with decent performance and became the biggest selling 9-pin printer in Europe.

Fair enough, but what has that to do with the price of cheese you may ask, and quite rightly too. Well, good old Citizen, flushed by the success of the 120-D has launched something of a successor, the Swift 24, which it hopes will owning the 24-pin market in a similar manner to the 120-D and the 9-pin market.

The Swift 24 offers the reasonably low price of £189 ex VAT and the power of a 24-pin printer, plus the convenience of no dip switches. Interested? Then read on.

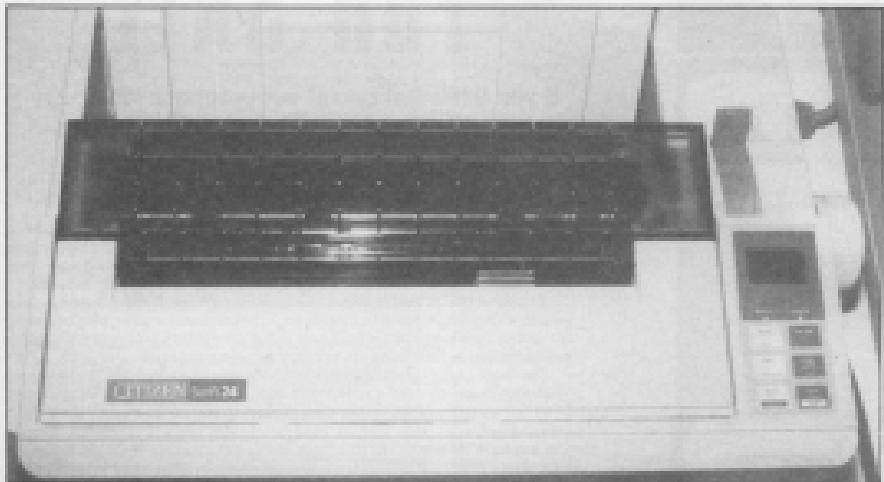
Amongst the many claims made for the Swift 24 by Citizen, is one of sturdiness. Ohay, let's see I thought, and dropped it on the floor from a foot. Apart from the top flapping around like a fish in the sea outside Salford, the Swift was unharmed by its flying lesson. But then again, who has their printer situated only a foot off the floor? I tried two feet, and was rewarded with an unendurable party of plastic whizzing past my ear as the shock waves set alarm bells ringing all over North London. The printer wasn't too happy with this treatment, and had to be calmed down and reassured before it was up to printing out this article. I still don't know where that piece of plastic came from.

The Swift has a two-year warranty from Citizen, as long as they don't catch you dropping it on the floor, and the print head is guaranteed to give you 200 million dots before dying, while the rest of the moving intervals will last approximately 4000 hours before ceasing to function. As long as you aren't rough with it, it should last five years with ease.

There's a tractor feed and an excellent single sheet paper feeder, and the printer is equipped with a Centronics parallel interface as standard with a serial RS232C as an optional extra.

Printing speed is another great claim in the advertising market, but at only 182 characters per second (at 100%) in draft mode its performance is only average against 9-pin printers in this price range. That's the fastest it goes, too; should also take into consideration 100 CPS (at 100% in draft), and more.

Since the time of, later than a
speaking parrot?



Bottom: Don't press the button that says
start, double click and Triple Click.

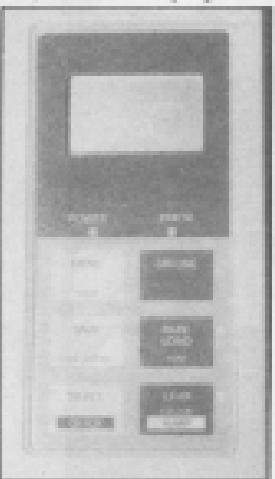
importantly 32 CPS at 10cpi and 64 CPS at 12cpi in NLQ mode. The NLQ fonts are presumably why you want this printer, otherwise I assume you are wasting your money. An 8K buffer helps the printing process, and a snappy little device tells you how much of the buffer is full of incoming data.

The built-in fonts are Times Roman (okay), Courier (every good), Helvetica (some serif and excellent) and Prestige Elite (horrible). There is a cutaway section of the printer where optional font cards can be plugged in to extend the range, and of course your software may be able to create additional fonts anyway. The dot density is 120 DPP in draft mode, and 300 DPP for PCL and graphics modes.

Instead of dip switches, the Swift has a two-line, eight character LCD, and eight button keypad. This can be used to alter all sorts of printer functions including character set (from UK to Latin American), page layout (one and two per inch), printer emulation (Epson LQ850, IBM Proprinter X24 and NEC P50), font, colour or standard ribbon, pitch (proportional and up to 2030DPI), and interface options.

The control panel also allows you access to four menus, which contain preselected settings. You can redefine any of these and save them into the Smith's electronically erasable programmable read only memory (EEPROM), so that when the printer is next switched on, your settings are automatically loaded as the default.

The *Entom Swift 24* is an excellent printer, offering the power of 24-pin printing, but with a low cost, considerable ease of use, a resilience to the sort of damage the average can inflict upon it, and it's backed up by a two-year warranty. If *Confederes* was alive today, the *Swift* could have saved the annotations of music, an awful lot of hand work.



Democracy

SOFTWARE FOR SALE

If you think that one of our programs looks very interesting, but you can't afford the time to type it in, then our software service will help you out.

It's three o'clock in the morning. You sit at the computer keyboard having just finished a marathon typing session extracting one of the superb programs from *Your Computer*. Your fingers reach for the keyboard and press the letters R, U and N. You press RETURN, sit back and nothing happens.

Everyone has probably faced this problem. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

The *Your Computer* Software Service makes available all of the programs from each issue on both cassette and disk at a price of £8.00 for disk and £4.00 for cassette. None of the documentation for the programs is supplied with the software since it is all available in the relevant magazine. Should you not have the magazine then back issues are available from the following address:

INFORNET LTD, 3 River Park Estate, Berkhamsted, Herts, HP4 3HE.
Tel (04427) 79661

Please contact this address for prices and availability.

The Disk

Programs on the disk will also be supplied as totally working versions, i.e. where possible we will not use Basic Loaders thus making use of the programs much easier. Unfortunately at the moment we cannot duplicate C16 and Plus/4 cassettes. However programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strap containing the article type, C16 Program etc. So that you can see which programs are available on which format, you will also find a couple of symbols after this strap. The symbols have the following meaning:



This symbol means that the program is available on cassette.



These programs are available on disk.

Please Note

Since the programs supplied on cassettes are total working version of the program, we do not put disk-only programs on tape. That is no sense in placing a program that expects to be reading from disk on to tape.

AUGUST 1918

REASONING ON THE E28 - Programs for the first part of our expert system series (E28 [Part 1/2]).

THESAURUS - Build up your own CM thesaurus with this program for disk or tape.

PRINTS ON A MAP - Generating maps in your GIS

ADDRESS BOOK - A contact-based address book for C64 users

DISK SCRAMBLER - Protect your disk data from prying eyes! (CD4 disk).

Sample Code

DETAILED PLAN

Table 1: Summary of Results

SEPTEMBER 1969

14500 AND 10000 AMES - Tutorial into mathematical finance

REFERENCES ON THE 129-CONTINUATION ELEMENT

1979-81 - Estimated 1981-82 approach for 1981-82 year.

ENTER CODE

DISK RESPONSE DATA

OCTOBER 1988

REASONING ON THE 128 – Turn your 128 into an expert system (part 2)

FLASH UPDATES - These will be for your **Big Box**.

FILE STANDARDS - The proper use of a program such

FLASH 10-810 - You just can't keep colored-in bordering borders.

PEPSI 100% - You name PEPSI and they...

WATER CONSERVATION

WHAT THIS REPORT SHOWS

Cassettes or disks are available from March 1986. Please ring the editorial office (0181 2069311) for details of these.

TREBLE

Champions

A STUNNINGLY REALISTIC FOOTBALL STRATEGY GAME



THREE CONVERGENCE

Can T.O.P. Win the League Championship, the F.A. Cup and the League Cup?

An Amazing range of Realistic Features and Superb Gameplay combine to create all the Tension, Drama and Excitement of Football Management as you compete for Soccer Glory and -

AN OUTSTANDING RANGE OF FEATURES

- **Introduction of AI**:
The process of introducing AI into business.
- **AI in Business**:
The use of AI to support business operations.
- **AI in Marketing**:
The use of AI to support marketing activities.
- **AI in Customer Service**:
The use of AI to support customer service.
- **AI in Manufacturing**:
The use of AI to support manufacturing processes.
- **AI in Healthcare**:
The use of AI to support healthcare operations.
- **AI in Finance**:
The use of AI to support financial operations.
- **AI in Retail**:
The use of AI to support retail operations.
- **AI in Agriculture**:
The use of AI to support agricultural operations.
- **AI in Transportation**:
The use of AI to support transportation operations.
- **AI in Energy**:
The use of AI to support energy operations.
- **AI in Manufacturing**:
The use of AI to support manufacturing processes.
- **AI in Healthcare**:
The use of AI to support healthcare operations.
- **AI in Finance**:
The use of AI to support financial operations.
- **AI in Retail**:
The use of AI to support retail operations.
- **AI in Agriculture**:
The use of AI to support agricultural operations.
- **AI in Transportation**:
The use of AI to support transportation operations.
- **AI in Energy**:
The use of AI to support energy operations.

- Many countries have a central bank that coordinates and oversees the banking system.
- Central banks are responsible for monetary policy.
- Countries, including the United States, provide central banking services.
- The Federal Reserve System is the central bank of the United States.
- The Federal Reserve System is responsible for monetary policy.
- The Federal Reserve System is responsible for supervising and regulating the banking system.
- The Federal Reserve System is responsible for providing services to the banking system.
- The Federal Reserve System is responsible for providing services to the public.
- The Federal Reserve System is responsible for providing services to the government.
- The Federal Reserve System is responsible for providing services to the economy.



ANSWER PAPER OF THE EXAMINATION

AVAILABLE NOW FROM RETAIL OUTLETS
AND BY MAIL, ORDER FROM:

© 2000 by Kluwer Academic Publishers. Printed in the Netherlands.

These make your Changes or Fixes available, regardless of
Changes between your application and another. Just make
the correct changes there. No need to recompile or

2025 RELEASE UNDER E.O. 14176

EDITOR'S NOTE

Due to a production error in last month's *Four Computers*, the review of Challenge Software's *Treble Champion* carried the wrong INFO Box. Unfortunately the score presented did not also give the game the rating that it deserved.

My son the Editorial team apologize to Challenge Software, and the readers who may have been confused and we feel that it is only right to print the correct INFO box here.

INFO

Gameplay: 91%
Graphics: N/A%
Sound: N/A%
Availability: 91%
Overall: 91%

9998

"Treble Champions is possibly the most complete football strategy game to date..."

NEON ZONE

If it's Christmas, it must be coin-ops, coin-ops and more coin-ops. Well, that's what all the major publishers decided a few years back anyway and as the panthers have responded with snapping the wads of green stuff on customers all around the country, I guess they must have been right.

You can count the home format hits over the years — *Gauntlet*, *Deathra*, *Operation Wolf* — just three of the titles that took the elusive number one spot over the festive season on consecutive years, with a whole trail of others not far behind, all riding on the back of phenomenal arcade success.

Christmas '87, *Gauntlet*, the first four-player game ever in the arcades, '87 gave us *OutRun* the best and most glamorous driving game at the time and in '88 came *Op! Wolf* the not particularly ideologically sound electronic shooting gallery that virtually caused a world shortage of 10p pieces.

If you are an aficionado of coin-ops, you'll notice that there's a significant time lag between a machine hitting the arcades themselves and appearing on home formats. The reason for this is pretty obvious. Coin-op manufacturers program and manufacture the games to an almost finished state (bear in mind the scale of the operations here — Sega alone have a workforce of 400 engineers working on the next *Afterburner*), then offer the licence to program the game on home formats to the various software houses.

Decisions are made, deals — whereby obscenely large amounts of money change hands — are struck, and then the programming must begin. True, some poor son-of-a-bitch has now got to get down to the brass tacks of fitting some 4 meg extravagance into 512, 64... or if he's done something particularly nasty in a past life, 48K.

How the hell do you start doing that, you might ask yourself? Well, in some cases it's quite simple. You don't. Many of the present generation of games, particularly the Sega titles, work by throwing tonnes of hardware at the thing, pre-stuffing loads of sprites and just (just!) blitting them up on the screen very, very fast. And you can do that on any affordable home machine available today. So what do you do?

Some conversion houses take the view that if you can't do it properly, you don't do it at all and turn such work down. Others, bear in mind that there is a fair bit of cash being flung around for these jobs, not to mention possible prestige, set to work on cutting the original game down like a crazed plastic surgeon, so it can work on common home machines, after a fashion.

In the days when all that was in an arcade machine was a 6809 processor, 1024 of memory and a sound chip, true coin-op conversions were possible. Now, the coin-op hardware has accelerated away into the distance, way out of reach of even the high end home machines, like the Amiga. What you get now are coin-op fakesakes and anyone that thinks you get the same as the original deserves to get disappointed!

Maybe this is one of the reasons that coin-ops seem to be slightly less prevalent this year than previously. Additions are at the same as ever though. Last year it was *Afterburner* and *H-Type* — this year it's *Altered Beast* and a very ambitious conversion of *Power Drift*.

Power Drift in particular is going to pose very, very significant problems, being a selection of mindbending fast sports routines. Cut down the colour, take out the memory, is it really going to be the same kind of experience as the original which was primarily designed to be a坐-in, throw you around, assault you with ear-shattering decibels job? I think not, Brian.

How about US Gold, who must be pleased about failing to make the hat-trick with *Thunderblade* last time around? Only *Capcom* cities on release this year with a credible version of *Strider* hitting the streets about now and *Ghosts 'n' Goblins* coming in as the run up to December. There's a near perfect version of this on the Sega Mega-Drive console, so this is certainly a possibility in the authenticity stakes, but the gameplay is a bit unconvincing and too similar to the original *Ghosts 'n' Goblins* that did so well for *Elite* way back. But maybe the panthers will be rushing back for more of the same.

Perhaps it's *Ghosts again* who are in with the best sticks at the top spot, as *Chase HQ* made a good enough impression in the arcades and, although challenging to write, is basically a driving game where you can bash into the opposition. Got to be a contender along with *Op! Wolf* follow up, *Operation*

ARCADeS ■

Christmas is coming — and that means the start of the coin-op avalanche in the shops.

John Cook takes a look back at the arcade hits this year and wonders how the hell you fit a 4 Megabyte extravaganza into 512, let alone 64K!





Thunderbolt and Cabal

But in the Dark Horse department, you have to consider Dynamix who'll be releasing on all formats the fugitive, *Hand Driver*, from Atari Games. Now this title was the first coin-op to use filled polygon graphics (like *Castlevania*) — not counting the weird *I-Robot* — and is a great game to boot, boot.

Due to some dodgy programming statistics, the coin-op hasn't yet got a startlingly fast frame rate and the 16-Bit version should come out about the same, with the 8-Bits suffering but still, probably acceptable, and in trendy 3-D too. That's my tip for the top for this Christy, for what it's worth. But what about next year... for now is the time that next year's coin-op conversion lot start making their way into the arcades.

Driving games have, again, been making the arcade operators feel warm inside as the pennies drop merrily into the coin slots. The award for most technically advanced has to go to *Minigolf Man* from a company called Namco. Like *Hand Driver* this one is a filled poly game but with a difference. It's fast. Very, very fast with some heartbreakingly beautiful touches, such as fading the position of background objects up as they proceed towards you. Just like in real life.

It's a race game *par excellence* as well with smooth and quick opposition plus car handling you'd normally dream about. In two modes of difficulty — hard and bloody impossible — there's plenty of life in the thing, so get yourself along to an arcade and try it out now and good luck to the guy that has to put it on home format.

Sage is probably the single most successful arcade company. Its "I've got a hardware expanded sprite routine and I'm going to use it" approach has spawned many hits, along with some sophisticated sit-in units. Its latest is also its best — *Super Monaco Grand Prix*.

SMGP is also a racing game — but with a different breed compared to something like *Wing Commander*. If the latter were an elegant gyroscope, *SMGP* would be Mike Tyson. The thing impresses by sheer power of volume and speed and is guaranteed to leave you breathless and wanting more! Can't see the software houses turning this one down.

Capcom has come up with a new hardware configuration called its CP System, based on a couple of super custom chips that were developed in-house.

This board can obviously do great things — *Street Fighter* and the never *Wingon* and *Dynasty Warri* being three of them — but the feeling is that the best is yet to come.

In the last set-up stakes, the sprites are becoming bigger and bigger — in games such as *Violent Fight* and *Data East's* fantasy combat book, but nothing new beyond that.

After the success of *Op. Wolf*, there have been many similarly inspired games, the white knuckled *Mechanised Attack* from SNE, *Cabal* and the latest, *Dynamite Duke*. The only surprising thing was that the official follow-up from Taito, the two player *Operation Thunderbolt* was such flop. Overall, however, '89 has established that the common or garden hardware used in the everyday PCB games has vastly outstripped the capability of the most popular home computers.

A simple game such as SNE's *Prehistoric Man* has so much going on, in so many colours, even that would be difficult to undertake. The pace of technological change — the one that separated home computers and video games almost simultaneously — is so fast that in coin-op terms, the machine you bought last year, let alone five years ago, is now obsolete in arcade terms.

Which begs the question — if that is the case, what will we all be playing in two Christmases time? And on what machine? Enter stage left a Commodore games console, maybe? Well, whatever, keep that joystick handy, Player One, and give those aliens hell! Until then, if you have been, ta very much... Kick out.



HOLMESOFT

MAIL ORDER DIVISION OF ESTABLISHED RETAILER

DEPARTMENT FC
ALDERSWOOD
CENTRE
REEDLEY, DUDLEY
WV10 8LW
01384 570077
(01384 570077)

COMPUTER SOFTWARE
PERIPHERALS &
COMPUTER HARDWARE
PC'S & MACS
DATA PROCESSING
HOME COMPUTERS
HOLMESOFT
(01384 570077)

PC'S & MACS
DATA PROCESSING
HOME COMPUTERS
HOLMESOFT
HOME DAY
DISPATCH

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

COMMODORE
Disk User

FOR C64 AND C128 USERS

Alright then! You've bought your C64 or C128, Disk Drive and maybe a Printer and Colour Monitor. So what are you going to do now?

Are you content to simply use all this expensive equipment to play the odd game, or, do you want more from your computer system? If the answer is YES then this is for you.

Commodore Disk User (CDU) is a magazine packed with programs, features and reviews for the less games-only orientated C64/C128 user. Sit down and ask yourself these simple questions:

Do I want my programs ready to use without typing in endless listings?

Do I want quality software that serves a useful purpose?

Do I want the occassional good game to play?

Do I want interesting and in depth features on programming techniques?

Do I want a FREE disk every month?

If you can answer YES to any of these questions then CDU is for you.

CDU is produced with the serious user in mind. On the front of each issue is a disk that has at least 10 programs on it.

You can expect to see: Sprite Editors; Machine-Code Monitors; Assemblers; Graphics Packages; Disk Utilities; Business Packages; Games and much more. Occasional programs on such things as: Programming Sound and Music, Rasters, Interrupts and Extended Basics.

So 'Come on Down' the price is right! Take a step in the right direction and subscribe to COMMODORE DISK USER.

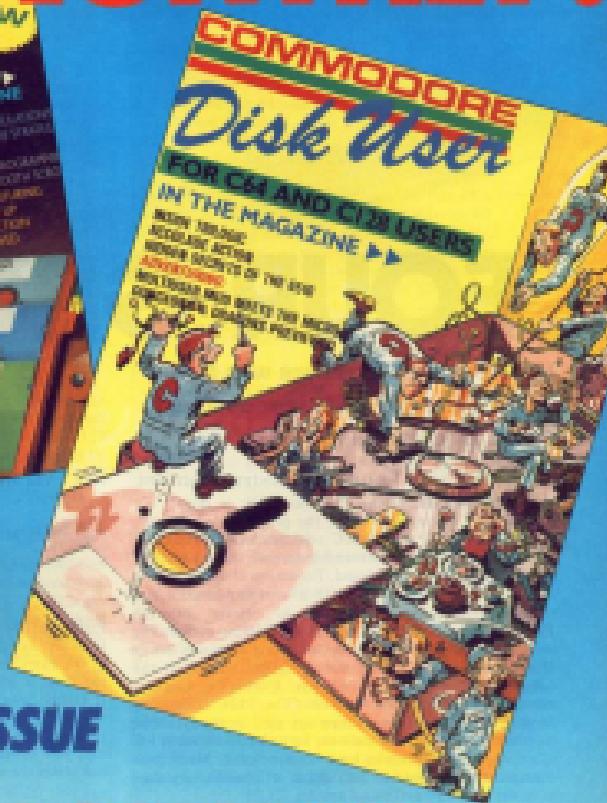
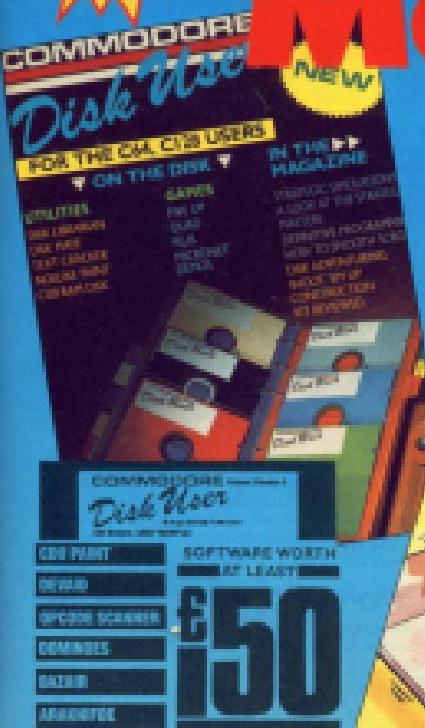
**GREAT
NEWS**

COMMODORE

Disk User

NEWS

**NOW
MONTHLY!**



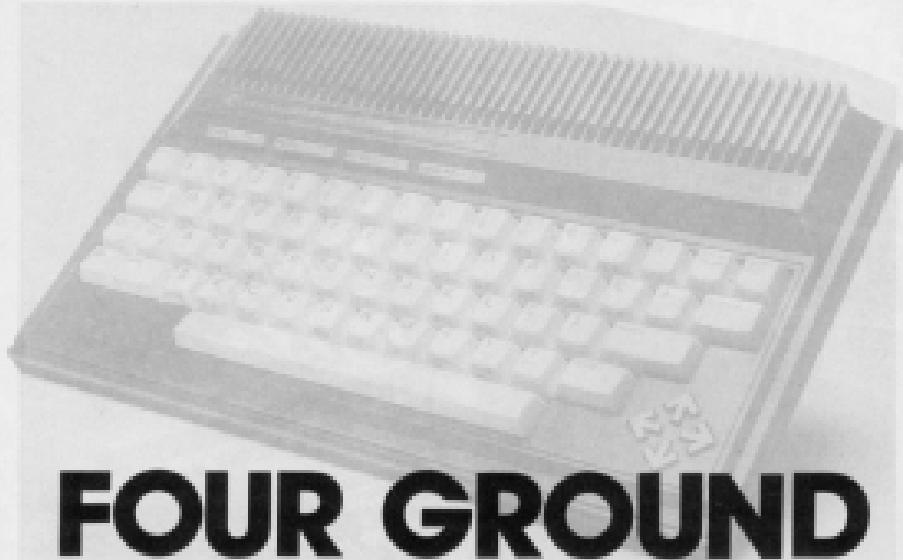
**NOVEMBER ISSUE
ON SALE**

OCTOBER 20TH

COMMODORE DISK USER is what you have been waiting for — take out a subscription TODAY!

COMMODORE DISK USER IS A LOT MORE THAN JUST ANOTHER COMPUTER MAGAZINE. EVERY ISSUE CARRIES A DISKETTE CONTAINING MORE THAN £50 WORTH OF SOFTWARE RANGING FROM SERIOUS PROGRAMMING UTILITIES TO ARCADE GAMES. THERE ARE PLENTY OF COMMODORE MAGAZINES ON THE MARKET, BUT WE BELIEVE THAT THIS IS THE FIRST TO CATER FOR DISK USERS OF ALL AGES AND TASTES.

Get in on the action



FOUR GROUND

Mark Everingham is back with the concluding part of his Introduction to Plus/4 Machine Language

7501 Machine Language Instruction Set

The instruction set of the 7501 processor is simply the collection of commands which the processor can understand, just like all the BASIC commands PRINT, LIST etc. What follows is a list of commands, each three characters long and requiring one operand. The action of the command is shown in symbolic form, with a brief explanation and example. All the examples can be entered into the TEDMON monitor using the method explained earlier. The valid addressing modes are shown along with the listing of any of the flags of the status register altered by the command. The meaning of each addressing mode is explained in the section entitled "The 7501 Addressing Modes". The conventions below are used in the symbolic representation of the operations. In order to conserve on space, only short examples have been included. Many such examples can be found in back issues of *Four Computers* or in the future *Four Ground* articles.

"#n" - Hex digit of number "M" - A memory/immediate
 "A" - The Accumulator "N" - The Negative Flag
 "B" - The BPL Flag "O" - The Overflow Flag
 "C" - The Carry Flag "X" - The X Register
 "D" - The Decimal Flag "Y" - the Y Register
 "I" - Interrupt Enable Flag "Z" - The Zero Flag

"[a]" - [b] - Bits: Bit move from [a] to [b]
 "[a]" - [b] - Bits: Bit move from [b] to [a]
 "[a]" - Bits: (0-7) in a byte

ADC (Add With Carry)

Operation: A=M+C → A, Flags Altered: N,Z,C,V

Addressing Modes: ADC #*hh* ADC S*hh* ADC S*hh,X*
 ADC S*hh,Y* ADC ADC (S*hh*,Y)
 (S*hh*,X)

Function: The ADC instruction is used to add two numbers together, using the accumulator. The operand + the setting of the carry flag (0 or 1) is added to the accumulator and the result is left in the accumulator. Any carry necessary is shifted into the carry flag. If the result is > 127 then the N flag is set. If the result is zero, Z is set. If the result is < -127 or < -128, the V flag is set. Normally prior to use, you should clear the carry flag. The example below calculates the sum of 16 and 48.

Example: 4800 LDA # 10 "Load A with value 10
 CLC "Clear Carry flag for addition
 ADC # 48 "Add value 48
 RER "Return to TEDMON

AND (Perform Logical AND On Accumulator)

Operation: A AND M → A, Flags Altered: N,Z

Addressing Modes: AND #*hh* AND S*hh* AND AND
 S*hh,X* AND S*hh,Y* AND AND
 (S*hh,X*) (S*hh,Y*) (S*hh,X*) (S*hh,Y*)

Function: Those bits which are both set to 1 in the accumulator and in the operand are left as 1 in the result. All other bits are set to zero. If the result is >127, N is set and if the result is zero, Z is set. The example below sets alternate bits in the byte 255 to zero.

Example: 4000 LDA #255 "Load A with value 255
 AND #1AA "And A with value 170
(alternate bits)
 BRK "Return to TEDMON

ASL (Arithmetic Shift Left)

Operation: C-[T...0]-0 Flags Altered: N,Z,C

Addressing: ASL, ASL \$AA ASL \$AA,X ASL \$AA,Y
Mode: ASL \$AA,X

Function: All the bits within either the accumulator or a memory location are shifted left. Bit 7 is shifted into the carry flag and Bit 0 is replaced with zero. The N and Z flags are set as for the AND instruction. Shifting a byte left has the effect in memory of multiplying it by two. The example below calculates 1024.

Example: 4000 LDA #24 "Load A with value 24
 ASL "Shift A left (#1)
 ASL "Shift A left (#1)
 BRK "Return to TEDMON

BCC (Branch If Carry Flag Clear)

Operation: Branch B<0 Flags Altered: None

Addressing Mode: BCC \$0000

Function: The BCC instruction examines the status of the carry flag and if it is set to zero, jumps to the specified address. Note that the address must be within -128 or +127 of the current address. This is automatically checked by TEDMON. The below example adds two numbers and if a carry does not occur, stores the result on the screen.

Example: 4000 : LDA \$A2 "Load A from address \$A2
 (Clock)
 CLC "Clear carry flag for addition
 ADC \$A6 "Add value in address \$A6
 (Clock)
 BCC \$0000 "If carry clear, jump to \$0000
 BRK "If set, return to TEDMON
 STA \$0C00 "Store result on screen
 BRK "Return to TEDMON

BCS (Branch If Carry Flag Set)

Operation: Branch B>0 Flags Altered: None

Addressing Mode: BCS \$0000

The BCS does the opposite of the BCC instruction. It tests the carry flag and if it is set, jumps to the specified address. Again, this address must be within -128 and +127 of the current address. See the example for BCC, substituting BCS for BCC. This stores the result in the screen memory if a carry did occur.

BEQ (Branch If Equal To Zero)
Operation: Branch If Z=1 Flags Altered: None
Addressing Mode: BEQ \$0000

Function: The BEQ instruction jumps to the specified address if the last result set the Z flag because it was equal to zero. The example below adds the value in \$D00 to the value in \$D01 and if the result is zero (they are both zero) then it stores an " " symbol on the screen.

Example: 4000 LDA \$D0 "Load A with value in \$D0
 CLC "Clear carry flag for addition
 ADC \$D1 "Add value in \$D1
 BEQ \$0000 "If result is zero, jump to \$0000
 BRK "Result is not zero - Return to TEDMON
 STA \$0C00 "Store value 0 on screen
 BRK "Return to TEDMON

BIT (Test Bits In Memory)

Operation: [T]-1 of M → N, [0]-1 of M → Y Flags Altered: N,Z,Y A and M → Z

Addressing Mode: BIT \$0000 BIT \$0000

Function: The BIT instruction tests the bits of a memory location against the value in the accumulator. Bit 1 of the memory value is transferred to N and Bit 0 to Z. If the result of ANDing A with the memory location is zero, the Z flag is set. Neither the values in the accumulator or in the memory location are changed. The example below looks at the first character on the screen and if it is reversed, replaces it with a space.

Example: 4000 BIT \$0000 "Test bits of value on screen
 BMI \$0000 "If character is reversed, jump to \$0000
 BRK "Not reversed - return to TEDMON
 LDA # \$20 "Load A with space character
 STA \$0C00 "Store space character on screen
 BRK "Return to TEDMON

BMI (Branch On Minus)

Operation: Branch if N<1 Flags Altered: None

Addressing Mode: BMI \$0000

Function: The BMI instruction tests the setting of the N (minus) flag and if it is set, jumps to the specified address. The N flag is set by other instructions where the resultant value is > 127 (bit 7 set). As for all other branches, the destination address must be within -128 and +127 of the current address. The example below counts from 128 to 256 in steps of two in the accumulator.

Example: 4000 LDA # \$00 "Load A with value 0
 CLC "Clear carry flag for addition
 ADC # \$02 "Add 2 to accumulator
 BMI \$0002 "If result > 127 jump to \$0002
 (Loop)
 BRK "Result < 128 - return to TEDMON

BNE (Branch If Not Equal To Zero)

Operation: Branch If Z≠0 Registers Altered: None

Addressing Mode: BNE \$0000

Function: The BNE instruction does the opposite of the BEQ command, performing a jump if the last result was

■ FOUR GROUND

not zero. See the example for BRQ, replacing BRQ with BNE.

BRK (Break Interrupt)

Operation: PC = Stack, SR = Stack, I = B Flags: None Altered

Function: The BRK instruction, which needs no operand, forces a Break interrupt to occur. On the Plus/4, this causes control to be transferred to TEDMON. In fact, the instruction jumps to the address found in \$0100 like a JMP (\$0100) command. The PC and SR settings before the command may be pulled off the stack using PLA.

BVC (Branch If Overflow Flag Clear)

Operation: Branch If V=0 Flags Altered: None

Addressing Mode: BVC \$0000

Function: The BVC instruction tests the status of the V flag in the status register and if it is clear (no overflow has occurred), does a jump to the specified address. There is no equivalent BVS instruction. The example loops until Bit 6 of the value in \$A00 (Clock) is set.

Example: **A000 BVC \$D0** Test bit of value in \$D0
BVC \$0000 *Overflow (Bit 6) clear - loop to \$0000
BRK *Overflow set - Return to TEDMON

CLC (Clear Carry Flag)

Operation: 0=C Flags Altered: C

Function: The CLC instruction which requires no operand, sets the value in the carry flag of the status register to zero. This is most useful in preparation for an ADC instruction. See any of the examples using ADC for a demonstration. See any of the examples using ADC for a demonstration.

CLD (Clear Decimal Mode)

Operation: 0=D Flags Altered: D

Function: The CLD instruction resets the D flag to zero, taking the Plus/4 out of decimal mode back into standard binary arithmetic. For an explanation of the decimal mode, see the SED instruction.

CLI (Clear Interrupt Disable Flag)

Operation: 0=I Flags Altered: I

Function: The CLI instruction clears the I flag and thus enables IRQ interrupt. No other flag is affected. For an explanation of interrupts, see the relevant section.

CLV (Clear Overflow Flag)

Operation: 0=V Flags Altered: V

Function: The CLV instruction simply resets the overflow flag of the SR to zero. Although this is not of any use for arithmetic operations, it can be used to produce a JMP effect using a CLV followed by a BVC. The use of this is that such a command can work when it is placed anywhere in memory whereas a JMP instruction will only work in the address for which it was intended. The example below is equivalent to JMP \$0010.

Example: **A000 CLV** *Clear the Overflow flag
BVC \$0010 *Branch to \$0010

CMP (Compare Value Against Accumulator)

Operation: No Effect Flags Altered: N,Z,C

Addressing	CMP	CMP \$0b	CMP	CMP
Mode:	* \$0b	\$0b,X	\$0b,Y	\$0b,Y
	CMP	CMP	CMP	CMP
	\$0b,X	\$0b,Y	\$0b,X	\$0b,Y

Function: The CMP instruction is used to compare a value with the accumulator in order to find which is larger, smaller, or if they are the same. The bits of the status flag are set then.

IF A < Value THEN N=1, Z=0, C=0

IF A = Value THEN N=0, Z=1, C=1

IF A > Value THEN N=0, Z=0, C=1

Combinations of these values are also possible, for instance just testing C shows that A is greater than or equal to the operand value. Once the CMP instruction has been carried out, you can test the settings of the flags and perform a branch accordingly. The example adds three to the accumulator until the result reaches 160.

Example: **A000 LDAA # \$00** Load A with value 0
CLC *Clear carry flag for addition
ADC # \$03 *Add value 3 to accumulator
CMP # \$A0 *Compare A to value 160
BNC \$0002 *If A < 160 jump to \$0002
BRK *value > = 160 - return to TEDMON

CPX (Compare Value Against X)

Operation: No Effect Flags Altered: N,Z,C

Addressing	CPX # \$0b	CPX \$0b	CPX \$0b,Y
Mode:			

Function: The CPX instruction performs an identical operation to CMP except that the comparison takes place between the X register and the specified operand value. The settings of the flags are identical to those for CMP.

CPY (Compare Value Against Y)

Operation: No Effect Flags Altered: N,Z,C

Addressing	CPY # \$0b	CPY \$0b	CPY \$0b,Y
Mode:			

Function: The CPY instruction performs an identical operation to CMP except that the comparison takes place between the Y register and the specified operand value. The settings of the flags are identical to those for CMP.

DEC (Decrement Memory)

Operation: M-1 → M Flags Altered: N,Z

Addressing	DEC \$0b,X	DEC \$0b,Y	DEC \$0b,X
Mode:	\$0b	\$0b,Y	\$0b,X

Function: The DEC instruction is used to decrease the value stored in a memory location by 1, as if you had subtracted one from it. The result of the decrement is stored back in the same location. If the value is decremented past zero it becomes 255. The N flag is set if the result is > 127, and the Z flag set if the result after the decrement is zero. The instruction does not set the carry flag like ADC and SBC. The example below decreases the border value until it becomes black (value 0).

Example: 4000 DEC SFFF19 'Decrement border colour in SFFF19'

BNR 50000 'If result < 0 jump to 50000'
BRK 'Result = 0 - Return to TEDMON'

DCK (Decrement X)

Operation: $X-1 \rightarrow X$ Flags Altered: N,Z

Function: The value currently in the X register is decreased by one and returned to the X register. As for DEX, if the result goes below zero, it becomes 255. The N and Z flags are set in the same way as by the DEX command. The example below uses the DCK instructions to provide short pause in the running of a program.

Example: 4000 LDCK = 0 'Load X with initial value 0'
DEX 'Decrement value in X'
BNR 50000 'If result < 0 jump to 50000'
BRK 'Return = 0 - return to TEDMON'

DEY (Decrement Y)

Operation: $Y-1 \rightarrow Y$ Flags Altered: N,Z

Function: The DEY instruction performs an identical action to DEX, except that it operates on the Y index register. The example program for DEY will work equally well if LDY and DEY are replaced by LDY and DEY. Settings of flags are as for DEX.

DEOR (Perform Exclusive-Or On Accumulator)

Operation: A EOR M \rightarrow A Flags Altered: N,Z

Function: Those bits that are set to one in either the accumulator or the specified operand are set to one. Those bits which are both zero in accumulator and operand are set to zero. Those bits which are set to one in both accumulator and operand are set to zero. The result is left in the accumulator. If the result is > 127 , N is set, and if the result is zero, Z is set. The EOR instruction has the effect of inverting bits in a byte.

INC (Increment Memory)

Operation: M+1 \rightarrow M Flags Altered: N,Z

Addressing: INC Snn,X INC Snn,Y INC Snn,nn

Function: The INC instruction does the opposite of the DEX instruction, adding one to the value stored in a specified memory location. The result after the incrementation is stored back in the source location. If the value is incremented past 255, it returns to zero. The N flag is set if the result is < 127 , and the Z flag if the result is zero. The example below increments the border colour until it goes past 255 back to 0 (black).

Example: 4000 INC SFFF19 'Increment border colour in SFFF19'

BNR 50000 'If result < 0 jump to 50000'
BRK 'Result = 0 - return to TEDMON'

INX (Increment X)

Operation: $X+1 \rightarrow X$ Flags Altered: N,Z

Function: The INX instruction performs the same operation as INC except that it increments the X register rather than a memory location. The operation and flag settings other than this are identical to INC. The example below

increments the value originally in the X register until it reaches 192.

INX (Increment X)

Operation: $X+1 \rightarrow X$ Flags Altered: N,Z

CPX = 192 'Compare value in X to 192'
BNE 50000 'If X < 192 jump to 50000'
BRK 'X = 192 - return to TEDMON'

JMP (Jump To Location)

Operation: M \rightarrow PC Flags Altered: None

Function: The JMP instruction is used to change program flow to a specified address, like the BASIC GOTO command. None of the flag or register settings are changed. The jump is unconditional and is not limited to the -128, +127 range imposed on branch instructions. The example below just jumps back to itself for ever!

Example: 4000 JMP 50000 'Jump unconditionally to 50000'

JSR (Jump To Sub-Routine)

Operation: PC \rightarrow Stack, M \rightarrow PC Flags: None Altered:

Addressing Mode: JSR Snnh

Function: The JSR instruction is equivalent to the GOSUB command in BASIC. First the address to which control must return after the sub-routine is pushed onto the stack, and then a jump is carried out to the specified absolute address. When a subsequent RTS is found, the return address is popped back off the stack and jumped to. The example below calls a short routine to increment the border colour, then exits.

Example: 4000 JSR 50004 'Jump to sub-routine at 50004'
BRK 'Return to TEDMON'
INC SFFF19 'Sub-routine - increment border colour'
RTS 'Return from sub-routine'

LDA (Load Accumulator)

Operation: M \rightarrow A Flags Altered: N,Z and 255 set by extension

Example: 4000 LDA SFFF19 'Load A with background colour in SFFF19'
EOR # SFFF19 'Toggle every bit in byte'
STA SFFF19 'Store back in background colour'
JMP 50000 'Loop back to start'

Addressing Mode:

LDA # Snn	LDA Snn	LDA Snn,X
LDA Snnh	LDA Snnh,X	LDA Snnh,Y

LDA (Snn,X) LDA (Snn,Y)

Function: The LDA instruction is used to transfer a value to the accumulator. The specified operand value is loaded into the accumulator. The N flag is set if the value is < 127 , and Z is set if the value is zero. The source value is not altered. For example, see any of the example programs using the instruction.

LDX (Load X)

Operation: M \rightarrow X Flags Altered: N,Z

■ FOUR GROUND

Addressing Modes:

LDX #8h	LDX 8h	LDX 8h,Y
LDX #8h,X	LDA 8h,X	

LDX instruction is identical in use to LDA but the specified value is loaded into the X register.

LDY (Load, Y)

Operation: $A \leftarrow Y$ Flags Altered: N,Z

Addressing Modes:

LDY #8h	LDY 8h	LDY 8h,X
LDY #8h,X	LDA 8h,X	

Function: The LDY instruction is identical in use to LDA but the specified value is loaded into the Y index register.

LSR (Logical Shift Right)

Operation: $A = [1...0] \rightarrow C$ Flags Altered: N,Z,C

Addressing: LSR
Modes: 8h,X 8h,Y 8h,X,8h,X

Function: The LSR instruction performs the opposite of the ASL instruction. All the bits within the accumulator or memory location are shifted to the right. Bit 0 falls out into the carry flag, and bit 7 is replaced with zero. This has the effect of dividing a number by two with the binary carry bit being left in the carry flag. The example below calculates 192/2.

Example: 6000 LDA \$C0 Load A with value 192
LSR Shift A right/2
LSR Shift A right/2
LSR Shift A right/2
BPL Return to TEDMON

NOP (Null Operation)

Operation: None Flags Altered: None

Function: The NOP instruction simply does nothing - it does not affect any register or flag, and creates a time delay of 2 clock cycles. The instruction is often used either to give a space between sections of a program or to replace other instructions during debugging.

ORA (Perform Logical OR)

Operation: $A \text{ OR } M \rightarrow A$ Flags Altered: N,Z

Addressing: ORA
Modes: #8h ORA 8h ORA
#8h,X ORA 8h,X ORA
8h,Y ORA 8h,Y ORA

Function: All the bits which are set to zero in both accumulator and operand byte are set to zero. All bits which are set to one in either or both of the accumulator and operand are set to one. The result is left in the accumulator. The N and Z flags are set as for the AND instruction. The example below sets bit 0 of the character at the start of the screen, so reversing it.

Example: 6000 LDA \$0C00 Load accumulator from screen byte
ORA #80 Set bit 0 (OR with value 128)
STA \$0C00 Return byte to screen
BPL Return to TEDMON

PHA (Push Accumulator onto Stack)

Operation: $A \rightarrow \text{Stack}$ Flags Altered: None

Function: The PHA instruction is used to push the value currently in the accumulator onto the machine stack. Once there, it can be pulled back into the accumulator using PLA. For information on the stack, see the relevant section. The example below uses a push and then pull to set up the status register flags.

Example: 6000 PHA Push value in accumulator onto stack

PLA Pull value back of stack & set flags
BPL Return to TEDMON

PHP (Push Processor Status Register onto Stack)

Operation: SR \rightarrow Stack Flags Altered: None

Function: The PHP instruction does the same as the PHA instruction except that it is the status register which is pushed onto the stack rather than the accumulator. This is useful for two purposes: retaining the status register when calling a sub-routine and examining the status register by pushing PHP and pulling PLA.

PLA (Pull Accumulator From Stack)

Operation: Stack \rightarrow A Flags Altered: N,Z

Function: The PLA instruction pulls a value off the stack into the accumulator previously pushed on using the PHA instruction. The N and Z flags are set in accordance with the manner for the LDA instruction. For an example, see PBA.

ROL (Rotate Left)

Operation: $C = [1...0] \rightarrow C$ Flags Altered: N,Z,C

Addressing: ROL
Modes: 8h,X ROL 8h,X ROL
8h,Y ROL 8h,Y ROL

Function: Bit 7 of the accumulator or memory location is shifted into the carry flag. Bits 6 through 1 are shifted left one bit, and the carry flag before the operation is shifted into bit 0. The result is left either in the accumulator or in the memory location specified. If the result is > 127, N is set and if the result is zero, Z is set. The example below uses ROL and ASL to multiply a 16-bit number by two. The low byte (8hLL) is stored in \$D0 and the high byte (8hH) in \$D1.

Example: 6000 ASL \$D0 Shift low byte in \$D0 left (128)
ROL \$D1 Rotate high byte + carry left
BPL Return to TEDMON

ROL (Rotate Right)

Operation: $C \rightarrow [1...0] \rightarrow C$ Flags Altered: N,Z,C

Function: The ROR command does the opposite of ROL. Bit 0 is shifted into the carry flag. Bits 6 through 1 are shifted right one bit, and bit 7 is replaced by the carry flag prior to the operation. This can be used in conjunction with LSR to perform two-byte division by two. The example below divides the 16-bit value in \$D0 (low) and \$D1 (high) by two.

Example: **6000 LSR SDI**

'Shift high byte in SDI

RDR SDI

right 4 (2)

right 4 (2)

right 4 (2)

RTI (Return From Interrupt)

Operation: Stack → SP, Stack → PC Flags Altered: All

Addressing Mode: Implied

Function: The RTI instruction is used like RTS but to return control to the main program after an interrupt has been serviced. It is usually not used as the Plus/4 in the Plus/4's own interrupt handler is normally used. For more information, see the section on interrupts.

SBC (Set Carry Flag)

Operation: I → C Flags Altered: C

Function: The SBC instruction simply sets the carry flag in the status register to one. This is mostly used as preparation for the SBC instruction which requires the flag to be set before subtraction. See SBC for an example of its use.

SED (Set Decimal Mode)

Operation: I → D Flags Altered: D

Function: The SED instruction puts the 7501 processor in your Plus/4 into its decimal simulation or BCD (Binary Coded Decimal) mode. Normally, numbers are represented in binary, 8 bits holding a number up to 255. In decimal mode, each byte holds a number between 0 and 99. The numbers are represented by two hex digits: 000 is 99 decimal, 320 is 23 decimal and 547 is 87 decimal, digits over 99 not being allowed. The mode is of limited use. The only time I have used it myself is for an on-screen clock in my COMM32 program to display the "Year Commander" in the future. All the arithmetic (ADD and SBC) instructions work while in decimal mode. The example below goes decimal mode to add 23 and 54 together, giving a BCD result.

Example: 6000 SED

'Enable decimal mode

LDA #52 'Load A with BCD value 23

CLC 'Clear carry flag for addition

ADC #54 'Add BCD value 54 to A

BRK 'Return to TEDMON

SEI (Set Interrupt Disable Flag)

Operation: I → I Flags Altered: I

Function: The SEI instruction disables all IRQ interrupts by setting the interrupt disable flag in the status register. For a full discussion, see the section on interrupts.

TAX (Transfer Accumulator to X)

Operation: A → X Flags Altered: N,Z

Function: The TAX instruction simply moves the value in the accumulator into the X index register. The command also sets the N and Z registers accordingly, as if a LDA instruction had been performed. The example below pulls a value off the stack and transfers it to the X register. This is useful as there are no PLX, PLY instructions.

Example: 6000 PLA 'Pull byte off stack into accumulator

TAX 'Transfer value into X

BRK 'Return to TEDMON

TAY (Transfer Accumulator to Y)

Operation: A → Y Flags Altered: N,Z

Function: The TAY instruction works in exactly the same way as the TAX instruction, transferring the value in the accumulator into the Y register and setting the N and Z flags as for LDA. See TAX for an equivalent example.

TSX (Transfer Stack Pointer to X)

Operation: SP → X Flags Altered: N,Z

Function: As mentioned in the section on the stack, the pushing and pulling values on the stack is handled by an index register, the stack pointer (SP). The value in the stack pointer may be transferred into X using the TSX instruction. This provides an alternative means of accessing the stack, not needing the accumulator. The example below pushes a value onto the stack and then retrieves it in Y without using the accumulator.

Example: 6000 PHA 'Push accumulator onto stack

TSX 'Transfer SP to X register

INX 'Increment X register

LDY #100,X 'Pull Y from stack area

TSX 'Restore new SP setting

BRK 'Return to TEDMON

TXA (Transfer X To Accumulator)

Operation: X → A Flags Altered: N,Z

Altered:

Function: TXA works just like TAX except here the value stored in X is transferred to the accumulator instead of vice-versa. This can be especially useful when you want to store the value in X on the stack temporarily. The example below shows this.

Example: 6000 TXA 'Transfer value in X to accumulator

PHA 'Push value onto stack

BRK 'Returns to TEDMON

TSX (Transfer X To Stack Pointer)

Operation: X → SP Flags Altered: N,Z

Function: As the reverse of TSX, the value in the X index register is transferred to the stack pointer. This instruction should be used with care as incorrect setting of the stack pointer can cause a program to crash. For an example of correct use, see TSX.

TYA (Transfer Y To Accumulator)

Operation: Y → A Flags Altered: N,Z

Function: The TYA works just like TAX and TAY except that the value in the Y register is transferred to the accumulator. For an applicable example, see TAX and TAY.

7501 Addressing Modes

Most of the 7501 instructions just covered require a number as their operand. However, this number can take many forms, all of which lead to a single byte-long number. The way in which this final number is specified is called an addressing mode. In all operations except moves and jumps,

■ FOUR GROUND

the actual number is the contents of the address specified and not the address itself.

Implied Mode. With implied mode, no operand is specified. In a command like TPA or PIA, the location of the data byte is implied by the command. Also, commands like ASL or ROL can be used implied because shifting the operand changes the accumulator as the data location.

Immediate Mode ([#data]). In immediate mode, you specify the byte number directly rather than as an address.

Absolute Mode ([#addr]). In absolute mode, you simply specify an address of a memory location in the range 0-65535. The byte operand is then defined as the contents of the location you have specified.

Zero Page Mode ([#hi]). Zero page mode is identical to absolute mode except that the address you specify can only be between 0 and 255. The advantage of using this mode is that it is faster, and because the operand is 8-bit instead of 16-bit, the command takes up one byte less than when using absolute mode.

Relative Mode. Relative mode is that used for all branch instructions. The idea is that instead of specifying an absolute address, you provide an offset which is added to the current address to find the destination. The programmer never has to worry about how this offset is calculated, as TELMON does the calculation for you.

Indirect Mode ([#hi]). Indirect mode can only be used by one command, the JMP instruction. The 7341 looks at the absolute address you specify, and gets its jump address from there. The final address is stored with low byte (#LL) at the location specified, and the high byte (#HH) at the next location on.

Absolute Indirect Mode ([#hi,X]/[#hi,Y]). Absolute indirect mode calculates the address from which to get a byte by adding the value stored in the index register you specify to the base address. Thus if X is set to 34, LDA \$1200,X would get a byte from the address \$1234.

Zero Page Indirect Mode ([#hi,X]/[#hi,Y]). Zero Page indirect addressing is the same as absolute indirect except that the base address can be between 0 and 255 instead of 0 and 65535. The advantage is speed and compactness.

Indirect Indirect Mode ([#hi,Y]). Indirect indirect mode, sometimes called post-index, indirect mode is a little more complex. The address 34h is a zero page address (0-255). The base address is found in 34h (low byte) and 35h (high byte). From this on, the mode works like absolute indirect mode, 35h,Y with 35h being the address found at 34h in zero page. The Y register is added to the address to give the final address.

Indexed Indirect Mode ([#hi,X]). Indexed indirect mode, sometimes called pre-indexed indirect mode is a little strange. The address location is calculated thus: The zero page address 34h is added to the value in X. A base address is then loaded from the zero page address \$10h+X, and this forms the final 16-bit address. The mode seems of limited value, and I have never myself seen a program on any 7301-type processor using the mode!

Plus/4 ROM/RAM Paging Considerations For Data Transfer

Most computers, such as the BBC Micro have only one memory map. That is to say that RAM extends from \$0000-SFFFF and ROM from \$8000-\$FFFF. The Plus/4 however is more flexible than that. It has RAM from \$0000-SFFF, but the area from \$8000 onwards can either be RAM or ROM, giving the computer a full 64K of RAM. You can choose to have ROM at \$8000- by doing a STA

SPPME, and ROM by doing a STA SPPSF. All LDA, JSR etc. operations work on the current type of memory, ROM or RAM. Note that when using RAM at \$8000 onwards, you must disable interrupts first. The two programs below would appear at first to do the same thing, but the first loads the accumulator with a byte from ROM and the second from RAM, by using the STA instructions to page RAM or ROM in.

Program 1: SII
STA SPPSF
LDA \$8000
STA SPPSF
CLI
BRE

Program 2: SII
STA SPPSF
LDA \$8000
STA SPPSF
CLI
BRE

The simple method of choosing between ROM and RAM configurations makes the Plus/4 far more powerful than standard single memory map computers.

An Introduction To Plus/4 7501 Interrupts

Interrupts seem to baffle many people, but are really nothing to fear. Interrupts are just what they sound like - interrupts! You can program your Plus/4 so that whenever a specific event occurs, it interrupts the main program which is running, and says "Hang on a minute, I'm sorry to interrupt but I've got something important to do now so you'll have to wait until I've finished." Your program, called the interrupt handler can then do whatever it wants, then it returns control back to the main program which continues running as if nothing had happened.

There are several different events which can be used to generate interrupts, but here we will consider only the simplest, what are called raster interrupts. Normally, interrupts are used on the Plus/4 to keep the system clock read by TI and TBI running. Every time the screen on your TV or monitor is refreshed, an interrupt is generated. This occurs every 50th of a second in the UK. When this interrupt is generated, a JMP instruction is done: JMP (\$0014), \$0014 being a "vector" containing the address at which an interrupt handler can be found. Then, a routine in the ROM of your Plus/4 updates the clock, scans the keyboard and does miscellaneous other work. This means that you can get your own program to run at the same time as BASIC, by changing the address stored in \$0014. The program below sets up interrupts so that every 50th of a second, the border colour is incremented.

\$0000 SII	Disable interrupts for setup
LDA # \$0D	*Load A with low byte of \$000D
STA \$0014	*Store low byte in interrupt vector
LDA # \$0B	*Load A with high byte of \$000D
STA \$0015	*Store high byte in interrupt vector
CLI	*Return to BASIC
RTS	*Return to BASIC
PHP	Interrupt Handler - Push SR to stack
INC SPPFF	*Increment border colour
PLP	*Pull SR back of stack
JMP \$C00E	*Return to ROM interrupt handler

The program can be entered into TELMON and then

MAIL BAG

Put pen to paper and tell us your news, views and groans

Dear VC

I feel that I must write and express my views.

I purchased a C16 in 1984 and enjoyed the scope of this machine so much that the following year I bought a Plus/4, and am still impressed by its capabilities.

However, things might have been different if I had known at the time that both of these machines were only a stop-gap until the C16 and Amiga were developed more.

Although I am satisfied with the Plus/4, the software manufacturers appear not to be, support for the machine has been poor. It is picking up slowly now but the demand will never be as great as for the C16, because it lacks the numbers game.

So, why not help all those C16 and Plus/4 owners and compile a C16/Plus/4 edition of *Your Commodore*, you already have the material to hand, so why not re-cycle and up-date a selection of articles from the last five years' magazines. Your range of material must be vast, therefore this edition would appeal to all levels of operators.

Perhaps if you did run such an edition, you might influence some software manufacturers to think again - not such a bad idea!

E. Williams, St. Yarmouth, Norfolk.

I agree wholeheartedly with you. The Plus/4 is a very much underrated machine.

However, you cannot simply just produce a one off magazine like this every so often. In fact, we have already published the 'Your Commodore C16 and Plus/4 essential guide', back in 1988. On top of this, we have to also get no less than 2 disks of Plus/4 software available through our Readers Services department. The fact of the matter is this, like any other commodity, computers and computer software rely on supply and demand. I think that the following will clearly show what I mean.

At the moment, I have 114 reader submissions awaiting publication. This figure can be broken down as follows:

C16 = 110, C128 = 4, Amiga = 1, Plus/4 = 8, (C116 is void).

Dear VC

Could you, through your letters page, please help me solve one of life's great mysteries.

I have been searching this fair land of ours for some 18 months now, for a copy of the instruction manual for the *French Speech Cartridge*. Unfortunately, my efforts have proved fruitless. I am turning to you as a last resort. Could you please publish my plea for any help in this matter. Thank you.

Ken Cole, 30 Ramsgate House, Ramsgate Street, London SE16 3NZ

Thanks for the letter. Here goes...

Come on now you lot out there, someone must be able to help our Ken. Any help would be most appreciated. Please send all replies direct to Ken and not to Your Commodore. Thank you.

Dear VC

The reason for this letter is simple. I am a dedicated C16-Plus/4 user. Therefore, I have started a C16-Plus/4 club. The club intends to produce a monthly magazine which will include Program listings, Tips, Hints, Pokes and Reviews.

If there are any C16-Plus/4 owners that would like to join the club, they can write to me at the address below for details. I would like to add that any C16-Plus/4 user can join, no matter what age, sex, level of competence you are.

R. Robinson, 112 Cliff Road, Hornsea, North Humberside, HU18 1JE.

Thanks for your letter. It is nice to see that someone is actually doing something positive for a change. Good luck with the venture.

Dear VC

I was interested to read your review article about the LC-10C colour printer. I purchased one about a month ago. On arrival, a small part of the membrane key system was defaced, which was of course no matter to the control or print output. But after 1st colour printouts, two hex changes in colour, then 8.0 test ones (in Black), in NLQ, the NLQ failed to before draft quality! I therefore phoned Star and spoke to someone that sounded like a young girl assistant. I do not think she understood what I explained to her. I later spoke to someone that appeared to be more technically minded. We both agreed that the ribbon should last for more than three few printouts in NLQ, this was at 11:30 in the morning. At 11:45 the next morning a replacement colour ribbon arrived by post. Blows that for excellent after sales service?? (11 out of 10 to Star).

Turned back with my thanks, and the following suggestions:

1. The clear part of the plastic cover should be removable in addition to the rest of the plastic cover. This lets one use the print as it is made.
2. As far as I know, only fanfold paper can be scrolled back by the printer, to the top of the page. Single sheets do not obey the command. It should be made possible for single sheets.
3. I found out that if one pulls the release lever to use them "backs" the single sheet with the first sheet of fanfold paper, then the command for the printer to return both the single sheet together with the first sheet of the fanfold paper does work. The fanfold sheet is the first of several sheets of the fanfold stack. The purpose of all that?? One can print in more detailed graphics to print detail.
4. Bradley, Lancashire, Scotland.

Thank you for your comments regarding the review. I have tried your suggestion regarding the scrolling and it does appear to work just fine. Thanks for the tips.

All your C128 problems answered by resident boffin Stuart Cooke.

**R
E
N
C
O
R
N
E
R
T
2
8**

Cry for help

Dear 128 Corner,

I wonder if you or any of your readers can help me. When using my 128 (or C16) mode I am unable to display the Pound sterling sign on either the screen or the printer.

On pressing the POUND key I get a Hash sign (#) displayed instead. Just as if I had pressed the Hash key.

If you can solve this problem it would make life easier for me as at the moment when in C16/M I have to type £ounds sterling.

I use a Wordprocessor called VDE11 which I obtained from the Public domain, (I am writing this with it) which I think is very good, but would be even better if I could use the POUND sign.

I would also like to know if it is possible to change the Character colour from purple/blue to C16/M. Can I change the colour from within a profile sub file, as I have put this file on my C16/M boot disk so as to ask me the Date and time on booting C16/M.

Four Commodores is a brilliant Mag for Commodore users and the 128 Corner is a great idea, its encouraged me to write to you, and I hope you are able to keep it going.

Steve Travis,
Barnsley

After much experimenting in the office I'm afraid that we've come up with a blank regarding your pound sign. You can alter the decompiling of the KEYDMP program that is on your system disk to alter the code that any key on the keyboard prints. All we can suggest is that you change the character code that is assigned to the pound key to the same code that your printer expects for this character. You may find that the character does not appear as a pound on the screen, however as it will be a blank slash, but as long as your printer is set up to print a pound you should have no problems. See your printer manual for details on the code to use for a pound sign.

The same program KEYDMP can be used to set the screen colours upon boot up. Don't make changes to the system disk supplied with your C16, make them on a copy of it. KEYDMP has quite a large help file associated with it so you should have no problem using it.

If anyone dreams of a better way to allow Mr. Travis to have a pound sign please let us know.

Good to hear that you like the mag and C128 Corner. Remember keep writing in with any tips, hints, views on software and general queries and we'll do our best to keep this page both topical and informative.

Video Memory Expansion

Dear 128 Corner,

In the July issue of Four Commodore you mentioned the video memory expansion from ESSL that brings the video memory of the VDC up to 64 K and you said that you don't know an order company that sells such an expansion.

There is a company in Switzerland that sells such an expansion and also a special Basic that supports the new memory. With this Basic you will be able to reach a resolution of 720x720 pixels.

The program is called Graphic Booster and it is available with the memory expansion for the C128/D in the plastic case and without the expansion for the C128/D in the metal case. Contact Combo AG, Tuggenweg 3, CH-4500 Solothurn, Switzerland.

J. Bets
Igerkingen
W. Germany

It's good to see that Four Commodore reaches parts that other computer mags cannot reach!

Thanks for the info on the Swedish company, other readers may wish to contact them directly at the address quoted. I have written to the company myself and will let you know through these pages about prices and any other C128 goodness that they may have available as soon as I get a reply.

18 into 8 won't go

Dear 128 Corner,

I have just read your article C128 Corner in the recent edition of *Your Commodore*, and many congratulations you upon taking the initiative to set up a regular page for C128 users.

I have been a Commodore fan for a number of years, and earlier this year decided to upgrade from the old faithful C64 which had served its purpose well but was stretched to its limits. Although the Amiga and PC clones looked very attractive, the initial outlay for both hardware and software was prohibitive so the C128D was bought, and C64 sold, the prospect of wider horizons for myself, and lots of old C64 software to keep the kids happy. Then the bubble burst...where was all the software and support for this machine...oh dear I've bought a dead 1100 m.s.

However I can now say how pleased I am with the C128D, with real wordprocessing courtesy of SuperScript, and a machine which can offer the budding programmer 2 processors to play with. I have programmed now to the stage of writing small Z80 routines in 128 mode and transferring them over into CP/M, and feel that the opportunities for learning are once again wide open. OK, my pain with the PC clones can still boast massive memories, terrific speeds, and huge cans for software, but many admit when having seen the C128 in action that its presentation and results are virtually as good, but the C128 is far more versatile and offers the learner far greater scope.

Is it beyond the bounds of possibility for an MS-DOS emulator to be constructed with the Z80, or is it exclusive to the 16-bit machines?? I have looked at a few MS-DOS books for hints on how the system is put together but all that I can find is user instructions... no nitty gritty!. Does anyone have any thoughts on this??

I have tried CP/M software from Digital which has been written for a Rainbow computer but the 128 is unable to read the disks, should I be looking at modifying the disk parameter table or am I expecting the impossible here??, any hints??

I will certainly do my best to keep the comments flowing in to support C128 corner, and I hope many of my fellow C128 users will do the same, well done to YOUR COMMODORE and thanks for your support.

Chris Aley
Langstone

Given to despair you're a C128 fan like others. Here at this office we still think that much of the software written for the C128 is far superior to that written for the 16-bit machines like the PC or Amiga. If you're into using CP/M why not get writing in with tips and articles so I'm sure that CP/M is an ability of the C128 that most users never see.

I'm afraid that MS-DOS is designed for 16-bit computers and 8-bit like the C128 and you'll be unable to run MS-DOS on your machine.

As for your problem running CP/M software for the Rainbow, there are two versions of CP/M available, one that can run C128 as drives for 8-bit computers like the Rainbow for its 16-bitness. You can run software written for 16-bit machines on your C128, you just of course can't load any piece of software written for 8-bit CP/M.

Get In Touch

C128 Corner is a forum for all 128 users, if you have any commercial enquiries or questions do send them in. Without your contributions this 128 Corner will not be able to continue to come on, write us:

128 Corner

Your Commodore
Argus House
Boundary Way
Hemel Hempstead
Bucks
HP2 7ST

THE END IS NIGH

Bribe of the Month



The big wobbly jelly spider
- Virgin Mastertronic



Ad Manager in brain swap shock

Paul Kavanagh, Ad Manager on your big wobbly VC, recently underwent a painful operation which scientists have been trying to perform for years. He volunteered to have his brain swapped with that of a newborn chimp, when asked if it was a success he replied "Oooh! Ooooh! Ooooh!". Fellow staff members have noticed a dramatic improvement.

Christmas is coming

As a special competition for Public Relations people, VC has opened a Christmas bribe bag. In our January issue (which promises to be big and wobbly) there will be a longer table with a list of the top ten franchises received by the press date, and the companies who donated them.

Reader participation

If anybody discovers a small Commodore/computer related news item, or piccy that they think would fit a page of this calibre, send it in and we'll raise a prize for every one printed. Be careful though, we don't want anything that may raise an UP to gutter level.

Gazza signs for wrong game

Paul Gascoigne was already confused at the recent press launch in aid of his new game. He performed a photo shoot in what one might call a "soft" Batman costume for the docto photographers that arrived. This was either a very clever ploy by Ocean software's Gary Hailey, as he was modelling next year's Tottenham strip, I know which excuse I'll plump for.



The Computer Industry Karma Sutra



A ball in the hand is worth two in the bush



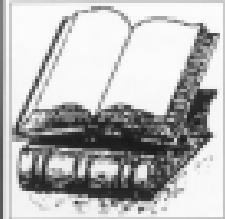
Freddy finally killed off

Not Bad US Gold does what no American teenager could, it has eliminated the master of disguise himself, Freddy Krueger. Although we had planned a large statue on Mt. Piranha, we have been informed that the computer game has been scrapped and there are no plans to resurrect it in the future. Mr. Evans you can take the mask off now. What do you mean "what mask?"

RAMSOFT OF ROCHDALE

(FORMERLY 16-BIT SOFTWARE)

AMIGA & ST BOOK AND HARDWARE SPECIALISTS



Amiga DOS Quick Reference Manual	£9.95
Motorola 68000 Programmers Ref Manual (Booklet)	£9.95
1001 Things To Do With Your Amiga (1st Ed)	£9.95
Amiga for Beginners	£12.95
Kickstart Guide to the Amiga (Amiga)	£12.95
Kids and the Amiga (Computer)	£12.95
Elementary Amiga BASIC (Computer) (2)	£12.95
Amiga Machine Language (Amiga) (2)	£12.95
Amiga Programmers Guide (Computer)	£14.95
Amiga DOS Reference Guide (Computer)	£14.95
Amiga Tricks and Tips (Amiga) (2)	£14.95
Inside Amiga Graphics (Computer) (2)	£14.95
Amiga C for Beginners (Amiga) (2)	£14.95
Amiga Applications (Computer) (2)	£14.95
First Book of the Amiga (Computer) (2)	£14.95
Amiga DOS - Inside & Out (Amiga) (2)	£14.95
Advanced Amiga BASIC (Computer) (2)	£14.95
Computer Manuals - Amiga Tech Gazette (Amiga - NEWT)	£14.95
Amiga Users Guide to Graphics, Sound, Telecom (Bantam)	£14.95
Becoming an Amiga Artist (Soft-Forwarder - 16bit)	£14.95
Amiga 3D Graphics Programming in BASIC (Amiga - NEWT) (2)	£17.95
Using Deluxe Paint II (Computer)	£17.95
Learning C - Graphics on Amiga & Amiga ST (Computer) (2)	£17.95
Amiga BASIC - Inside & Out (Amiga) (2)	£18.95
Amiga Microsoft BASIC Programmers Guide (Soft-Forwarder)	£18.95
Inside the Amiga with C (Sams)	£18.95
Amiga DOS Manual (Bantam)	£20.95
Programming for Amiga (Sams)	£20.95
Amiga Disk Drives - Inside & Out (Amiga) (2)	£24.95
Programmers Guide to the Amiga (Sams)	£24.95
Amiga Programmers Handbook (Sams)	£24.95
Amiga Programmers Handbook Vol 2 (Sams)	£24.95
Amiga ROM Kernel Ref Manual (Amiga - NEWT)	£29.95
Amiga C for Advanced Programmers (Amiga - NEWT)	£29.95
Amiga System Programmers Guide (Amiga) (2)	£29.95

PLEASE NOTE:

16-BIT SOFTWARE HAS NOW EXPLORDED A GREAT DEAL AND NOW
THROWN UNDER THE HEAVY SHED OF PARROT 1.
WE CAN NOW OFFER 48 HR DELIVERY ON ALL STOCK ITEMS.
IF THE ITEM YOU REQUIRE IS NOT ADVERTISED
PLEASE CALL US ON THE OFFICE NUMBER
WE CAN HAVE ALL ITEMS REQUESTED AT OUR SHOWROOM
WITHIN 24 HOURS (SUBJECT TO AVAILABILITY)

AMIGA HARDWARE OFFERS



• RAMSOFT PACK A

AMIGA 1000, 80K MEMORY, 1 MB INTERNAL DRIVE,
TV MODULATOR, KICKSTART 1.3, WORKBENCH 1.5,
TUTORIAL AND EXTRA DISKS, MOUSE AND MOUSE MAT,
10 TOP-FLO DISKS, CP10 BLANK DISKS AND CARDS

£259.00

• RAMSOFT PACK B

THIS PACKAGE AS ABOVE, PLUS 16-BIT STAR PACKED
VALUED AT OVER £200. PAY IN 4 PARTS AND PAY A
LIQUIDATED PENALTY, 10% ON 1ST, FINAL 10% ON

£379.00

• RAMSOFT PACK C

AMIGA 1000, KICKSTART 1.3, WORKBENCH 1.5, MOUSE MAT,
10 TOP-QUALITY F.D. GAMES, PHILIPS C1240 MONITOR

£545.99

ACCESSORIES

CURVIA 3.5 INCH DRIVE £29.95
KOPRI NAVIGATOR JOYSTICK £19.95
QUICKSHOT II TURBO JOYSTICK £29.95

PRINTERS

SENCORA EP-110 + LEAD	£109.95
CITIZEN 120-D + LEAD	£109.95
STAR LC 10 MONO + LEAD	£149.95
STAR LC 10 COLOUR + LEAD	£199.95

WHY NOT CALL IN OUR SHOWROOM WHERE EXPERIENCED AND HELPFUL STAFF CAN HELP SOLVE YOUR PROBLEMS!
OUR SHOWROOM ADDRESS IS: UNIT 1, 160 DRAKE STREET, ROCHDALE, LANC.

0161 425 1919 BETWEEN 9.00 AM & 5.30 PM

BOOK PROGRAMS ON CREDIT/DEBIT/ATM FOR ALL TITLES MARKED WITH A £10-£19.95

PLEASE NOTE: CODES CAN BE BOUGHT WITHOUT DISCOUNT. SEND CHEQUE/PO/ACCESS OR VISA DETAILS TO:
RAMSOFT, DEPT. P.O., UNIT 1, DRAKE HOUSE, 160 DRAKE STREET, ROCHDALE, LANCASHIRE, OL1 1FX

24 HOUR DISPATCH ON ALL STOCK ITEMS SUBJECT TO CHECKS, CLEARANCE

ALL OFFERS ARE SUBJECT TO AVAILABILITY AND MAY CHANGE AT ANY TIME. TA001



AMIGA

ATARI
ST

MICRO Music

IBM PC

APPLE
MAC

ELECTRONICA '89

VIEW

WE MEET A GUY
CALLED GERALD!

CLASH OF THE TITANS!

Cubase and Virtuoso Head-to-head Review

THE SPECIALIST
MUSIC MAGAZINE FOR
COMPUTER USERS

OCTOBER - NOVEMBER 1989
£1.75

Get Musical
with the
Archimedes

High Note
and Armadeus
Reviewed

ACORN

COMMODORE
64

AMSTRAD
CPC

ZX
SPECTRUM

Britain's First Music Magazine for the Computer User!

Featuring Reviews on Hardware & Software across all formats — Micro Music is the magazine the Computer Musician has been waiting for. Available 8th September from all good newsagents.